

FORM PTO-100  
(REV. 10-2000)

U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

ATTORNEY'S DOCKET NUMBER

T2146-906652

TRANSMITTAL LETTER TO THE UNITED STATES  
DESIGNATED/ELECTED OFFICE (DO/EO/US)  
CONCERNING A FILING UNDER 35 U.S.C. 371

U.S. APPLICATION NO. (If known, see 37 CFR 1.5)

09/700428

INTERNATIONAL APPLICATION NO.  
PCT/FR00/00625INTERNATIONAL FILING DATE  
15 MARCH 2000PRIORITY DATE CLAIMED  
15 MARCH 1999TITLE OF INVENTION SYSTEM FOR ACCESSING AN OBJECT USING A "WEB" BROWSER CO-OPERATING WITH  
A SMART CARD, AND ARCHITECTURE FOR IMPLEMENTING THE METHOD

APPLICANT(S) FOR DO/EO/US Pascal URIEN

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This is an express request to promptly begin national examination procedures (35 U.S.C. 371(f)).
4. ☒ The US has been elected by the expiration of 19 months from the priority date (PCT Article 31).
5. ☒ A copy of the International Application as filed (35 U.S.C. 371(c)(2))
  - a. ☒ is attached hereto (required only if not communicated by the International Bureau).
  - b. ☐ has been communicated by the International Bureau.
  - c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☒ An English language translation of the International Application as filed (35 U.S.C. 371(c)(3)).
7. ☐ Amendments to the claims of the International Application under PCT Article 19(35 U.S.C. 371(c)(3))
  - a. ☐ are attached hereto (required only if not communicated by the International Bureau).
  - b. ☐ have been communicated by the International Bureau.
  - c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
  - d. ☐ have not been made and will not be made.
8. ☐ An English language translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. ☒ An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).
10. ☐ An English language translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

**Items 11 to 16 below concern document(s) or information included:**

11. ☒ An Information Disclosure Statement under 37 CFR 1.97 and 1.98. w/references
12. ☒ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.  
to BULL CP8
13. ☒ A FIRST preliminary amendment.  
☐ A SECOND or SUBSEQUENT preliminary amendment.
14. ☐ A substitute specification.
15. ☒ A change of power of attorney and/or address letter.
16. ☒ Other items or information:  
Translated formal drawings (8)  
Verification of Translation  
Copies of PCT/RO/101; PCT/IB/301, 304, 308; DEMAND  
Proposed Drawing Corrections and 8 red-lined formal drawings

U.S. APPLICATION FOR PATENT (PCT/US-50)

0977700428

INTERNATIONAL APPLICATION NO  
PCT/FR00/00625ATTORNEY'S DOCKET NUMBER  
T2146-90665217. ☒ The following fees are submitted:**BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)):**

Neither international preliminary examination fee (37 CFR 1.482)  
nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO  
and International Search Report not prepared by the EPO or JPO ..... \$1000.00

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but  
USPTO but International Search Report prepared by the EPO or JPO: ..... \$860.00

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but  
international search fee (37 CFR 1.445(a)(2)) paid to USPTO ..... \$710.00

International preliminary examination fee paid to USPTO (37 CFR 1.482)  
but all claims did not satisfy provisions of PCT Article 33(1)-(4) ..... \$690.00

International preliminary examination fee paid to USPTO (37 CFR 1.482)  
and all claims satisfy provisions of PCT Article 33(1)-(4) ..... \$100.00

**ENTER APPROPRIATE BASIC FEE AMOUNT =**

\$ 860.00

Surcharge of \$130.00 for furnishing the oath or declaration later than ☐ 20 ☐ 30  
months from the earliest claimed priority date (37 CFR 1.492(e)).

\$

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE
Total claims	15 - 20 =	0	X \$18.00
Independent claims	5 - 3 =	2	X \$80.00

\$ 160.00

MULTIPLE DEPENDENT CLAIM(S) (if applicable)

+ \$270.00

\$

**TOTAL OF ABOVE CALCULATIONS =**

\$ 1,020.00

☐ Applicant claims small entity status. See 37 CFR 1.27. The fees indicated above  
are reduced by 1/2.

\$

**SUBTOTAL =**

\$ 1,020.00

Processing fee of \$130.00 for furnishing the English translation later than ☐ 20 ☐ 30  
months from the earliest claimed priority date (37 CFR 1.492(f)).

\$

**TOTAL NATIONAL FEE =**

\$ 1,020.00

Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be  
accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). \$40.00 per property +

\$ 40.00

**TOTAL FEES ENCLOSED =**

\$ 1,060.00

Amount to be

refunded: \$

charged: \$

a. ☐ A check in the amount of \$\_\_\_\_\_ to cover the above fees is enclosed.

b. ☒ Please charge my Deposit Account No. 501165 in the amount of \$1,060.00 to cover the above fees.  
A duplicate copy of this sheet is enclosed.

c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any  
overpayment to Deposit Account No. 501165. A duplicate copy of this sheet is enclosed.

**NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.**

SEND ALL CORRESPONDENCE TO

Edward J. Kondracki  
MILES & STOCKBRIDGE PC  
1751 Pinnacle Dr.  
McLean, VA 22102-3833

SIGNATURE

Edward J. Kondracki

NAME

20,604

REGISTRATION NUMBER

**IN THE UNITED STATES DESIGNATED/ELECTED OFFICE (D.O./E.O./US)**

Applicant: Pascal URIEN

International  
Application No.: PCT/FR00/00625

International  
Filing Date: 15 March 2000

U.S. Serial No.: To be Assigned

U.S. Filing Date: November 15, 2000

For: **SYSTEM FOR ACCESSING AN OBJECT USING A "WEB"  
BROWSER CO-OPERATING WITH A SMART CARD, AND  
ARCHITECTURE FOR IMPLEMENTING THE METHOD**

McLean, Virginia

**PRELIMINARY AMENDMENT**

Honorable Commissioner of Patents  
and Trademarks  
Washington, D.C. 20231

Sir:

Please amend the subject application, filed concurrently herewith, as indicated  
below:

**IN THE SPECIFICATION:**

After the title and before the first paragraph on page 1, insert the following:

-- FIELD OF THE INVENTION--;

Page 1, at line 8, before the paragraph beginning "Within the scope...", insert the  
following heading at the left hand margin:

--BACKGROUND DEFINITIONS--;

Page 1, line 23, after "includes", insert --the global network known as--

Page 1, at line 28, before the paragraph beginning "Hereinafter...", insert the following heading at the left-hand margin:

--DESCRIPTION OF RELATED ART--;

Page 1, line 31, delete "Internet" and substitute --internet--;

Page 2, line 5, after "architecture", insert --or system--;

Page 5, at line 8, and before the paragraph beginning "The object of the ...",

insert the following paragraph at the left-hand margin:

--SUMMARY OF THE INVENTION--;

Page 10, at line 2 and before the paragraph beginning "The invention will now...",

insert the following heading at the left hand margin:

--BRIEF DESCRIPTION OF THE DRAWINGS--;

Page 10, at line 23 and before the paragraph beginning " Before describing...",

insert the following heading at the left hand margin:

--DESCRIPTION OF THE PREFERRED EMBODIMENT(S)--;

Page 29, after line 21, insert the following new paragraph:

--While this invention has been described in conjunction with specific embodiments thereof, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art. Accordingly, the preferred embodiments of the invention as set forth herein, are intended to be illustrative, not limiting. Various changes may be made without departing from the true spirit and full scope of the invention as set forth herein and defined in the claims.--

**IN THE CLAIMS:**

Please cancel claims 1 - 15 in their entirety and without prejudice and substitute the following new claims:

17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2

--16. An embedded system adapted to cooperate with a network through a terminal, comprising a chip having an information processor and a memory for information storage,

- said memory storing at least one object file containing information associated with an object located in the network and making it possible to create an instance of this object;

- a first network interface adapted to cooperate with a second matching network interface located in the terminal, so that the embedded system constitutes an information server in the network; and

- a third object file interface for establishing correspondence between information passing through the first network interface and assigned to at least said object file, and information exchanged with said object file.

17. An embedded system according to claim 16, wherein the object file comprises a piece of autonomous software executable in browser software.

18. An embedded system according to claim 16, wherein said first network interface means adapted to cooperate with the matching network interface means located in the terminal, such that the embedded system behaves like a client capable of connecting to at least one server of the network.

19. A method for instantiating an object located in a network, characterized in that it uses an embedded system adapted to cooperate with a network through a terminal, comprising a chip having an information processor and a memory for information storage, the embedded system storing at least one object file containing information associated with an object located in the network for creating an instance of said object, and further comprising a first network interface designed to cooperate with a second matching network interface located in the terminal, such that the embedded system constitutes an information server in the network, and a third object file interface

adapted to establish a correspondence between information passing through the first network interface means and assigned to at least said object file, and information exchanged with said object file, the method comprising the steps

- establishing a list of the agents implemented; and
- for each agent, defining call arguments necessary to the agent so as to describe a set of sessions between agents using an object file..

20. A method according to claim 19, further comprising describing the opening of a session with another agent by a call argument.

21. A method according to claim 19 further comprising modifying the list of arguments used by a first agent by another agent.

22. A method for instantiating an object located in a network, characterized in that it uses an embedded system designed to cooperate with a network through a terminal, comprising a chip having an information processor and a memory for information storage, the embedded system storing at least one object file containing information associated with an object located in the network and for creating an instance of said object, and further comprising a first network interface designed to cooperate with a second matching network interface located in the terminal, such that the embedded system constitutes an information server in the network, and a third object file interface adapted to establish a correspondence between information passing through the first network interface and assigned to at least said object file and information exchanged with said object file, the method comprising the steps of:

- identification of an object file; and
- execution of this object file so as to implement sessions between agents described by an object file executed from the information server of the embedded system..

23. A method according to claim 22, wherein the object file is executed by

1 instantiating the first agent associated with the object file.

1 24. A method according to claim 22, wherein the object file is executed by  
2 instantiating one or more agents referenced by the object file.

1 25. A method for instantiating an object located in a network, characterized in  
2 that it uses an embedded system adapted to cooperate with a network through a  
3 terminal, comprising a chip having an information processor and a memory for  
4 information storage, the embedded system storing at least one object file containing  
5 information associated with an object located in a network and for creating an instance  
6 of said object, and further comprising a first network interface designed to cooperate  
7 with a second matching network interface located in the terminal, such that the  
8 embedded system constitutes an information server in the network, and a third object  
9 file interface adapted to establish a correspondence between information passing  
10 through the first network interface means and assigned to at least said object file and  
11 information exchanged with said object file, the method comprising the steps of:  
12 - loading an object file and a specific software capable of implementing it by  
13 browser software; and  
14 - execution of the specific software by the browser software so as to implement  
15 sessions between agents described by an object file executed from browser software..

1 26. A method according to claim 25, wherein the specific software is  
2 embodied in an interpreted language executable by the browser software.

1 27. A method according to claim 25, wherein an object file interpreter is  
2 embodied in the browser software.

1 28. A method for instantiating an object located in a network, characterized in  
2 that it uses an embedded system adapted to cooperate with a network through a  
3 terminal, comprising a chip having an information processor and a memory for



1 information storage, the embedded system storing at least one object file containing  
 2 information associated with an object located in the network for creating an instance of  
 3 said object, and further comprising a first network interface adapted to cooperate with a  
 4 second matching network interface located in the terminal, such that the embedded  
 5 system constitutes an information server in the network, and an object file interface  
 6 adapted to establish a correspondence between information passing through the first  
 7 network interface means and assigned to at least said object file, and information  
 8 exchanged with said object file, the method comprising the step of identifying, by  
 9 means of a universal resource identifier, a specific software implementing the browser  
 10 software so as to enable the embedded system to implement sessions between agents  
 11 described by an object file executed from browser software.

1 29. A method according to claim 28, wherein a universal resource identifier is  
 2 integrated into a hypertext document.

1 30. A method according to claim 28, wherein said specific software is loaded  
 2 by a method available in the browser software and deduced from the universal resource  
 3 identifier.--



## --ABSTRACT

The invention relates to a method and an architecture for securely accessing virtual objects (*Obv*) distributed in systems connected to the internet network (*Ri*), and for obtaining an instance of same. This access is performed via a smart card (2a), through a "web" browser (10). The terminal (1) and the smart card (2a) each comprise a specific protocol layer (13, 23a). The latter comprises intelligent agents (132, 232a<sub>1</sub>) for establishing two-way data exchange sessions, thereby allowing the smart card (2a) to have a "web" server functionality. The smart card (2a) also comprises intelligent agents, called script translators, and a virtual file management system (8) cooperating with a specialized script-translating intelligent agent (7). Each virtual object (*Obv*) is associated with a virtual file of the virtual file management system (8). The specialized intelligent agent (7) presents the browser (10) with a list of the accessible virtual objects (*Obv*) and generates methods for accessing these objects.--

**REMARKS**

This Preliminary Amendment is filed to insert headings to conform the application to U.S. practice, and to correct informalities in the specification, claims and abstract resulting from a literal translation of the French text.


Early action on the merits is earnestly solicited.

Respectfully submitted,

MILES & STOCKBRIDGE P.C.

Date: November 15, 2000

By:

  
Edward J. Kondracki  
Registration/No. 20,604

1751 Pinnacle Drive – Suite 500  
McLean, VA 22102-3833  
Tel.: 703/903-9000  
Fax: 703/610-8686

T2146-906652-US3815/BC(PCT)

**IN THE UNITED STATES DESIGNATED/ELECTED OFFICE (D.O./E.O./US)**

Applicant: Pascal URIEN

International  
Application No.: PCT/FR00/00625International  
Filing Date: 15 March 2000

U.S. Serial No.: To be Assigned

U.S. Filing Date: November 15, 2000

For: **SYSTEM FOR ACCESSING AN OBJECT USING A "WEB"  
BROWSER CO-OPERATING WITH A SMART CARD, AND  
ARCHITECTURE FOR IMPLEMENTING THE METHOD**

McLean, Virginia

**PROPOSED DRAWING CORRECTIONS**Hon. Commissioner of Patents and Trademarks  
Washington, D.C. 20231

Sir:

Applicant requests approval of the drawing corrections in Figs. 1 – 14 as  
shown in red on the attached eight (8) sheets.

The proposed corrections only comprise translating the French terms into  
English, labeling the blocks to correspond the drawings to the specification and

T2146-906652-US3815/BC(PCT)

claims, and removing the headings "1/8" to "8/8" to conform the drawings to U.S. practice.

Respectfully submitted,

MILES & STOCKBRIDGE P.C.

Date: November 15, 2000

By: 

Edward J. Kondracki  
Registration No. 20,604

1751 Pinnacle Drive – Suite 500  
McLean, VA 22102-3833  
Tel.: 703/903-9000  
Fax: 703/610-8686

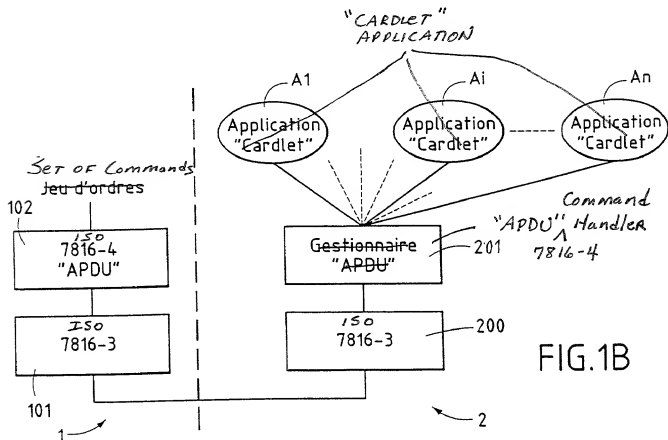
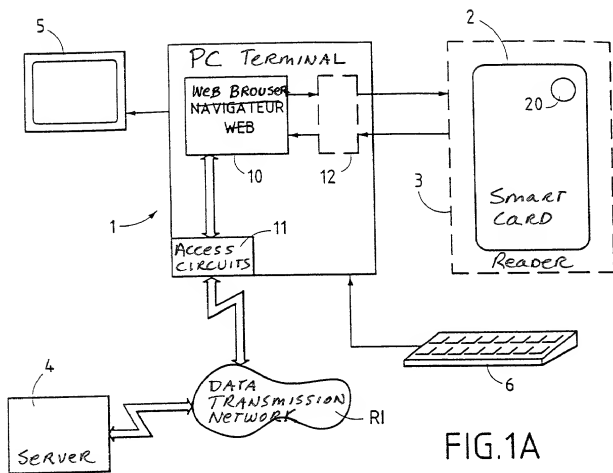
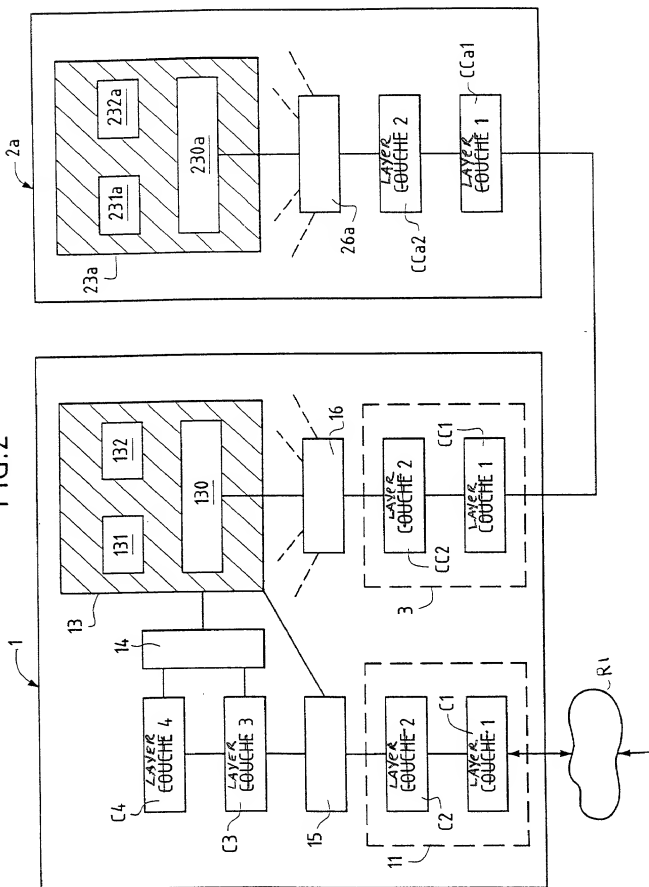


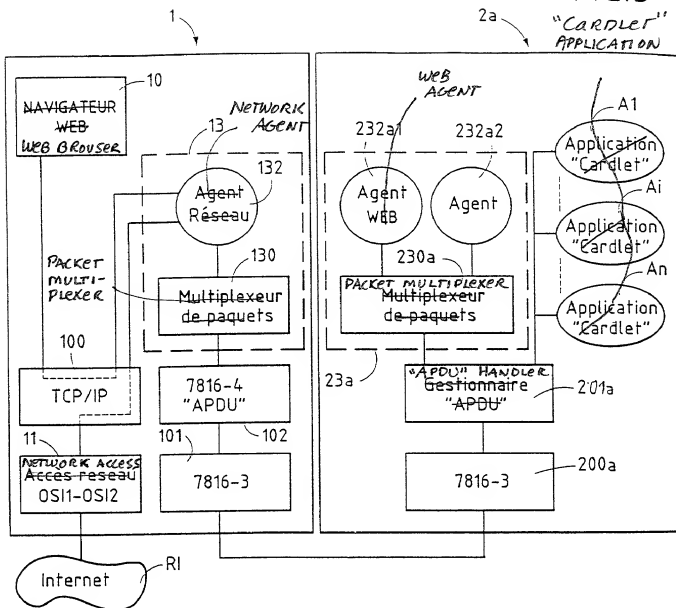
FIG. 2



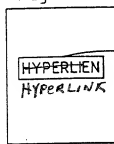


3/8

FIG.3



Html Page  
Page Htmlt



http://@Agent:AgentPort/MethodPDU?value=xx

Agent  
Serveur  
TCP/IP

232ap  
TCP/IP SERVER  
AGENT

1<sup>er</sup> Agent De la Liste 1<sup>st</sup> Agent on the  
de Structure d'Appel LIST OF "METHOD PDU"  
"MethodPDU" Call Structures

FIG.15

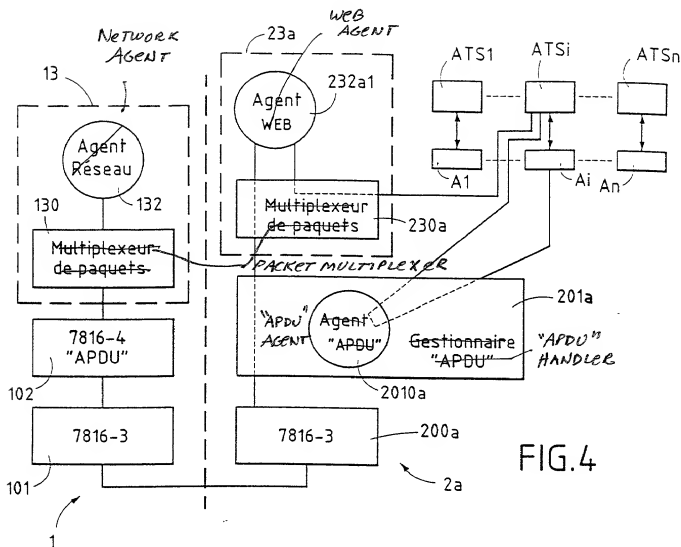


FIG. 4

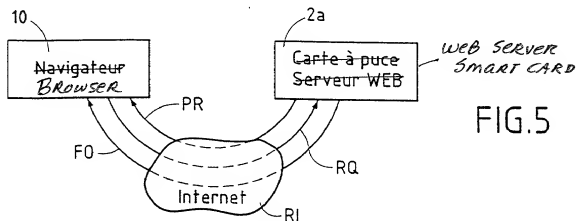


FIG. 5

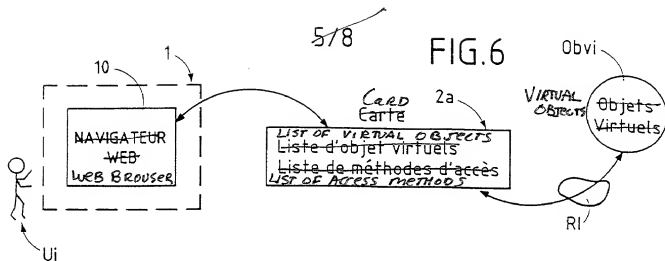
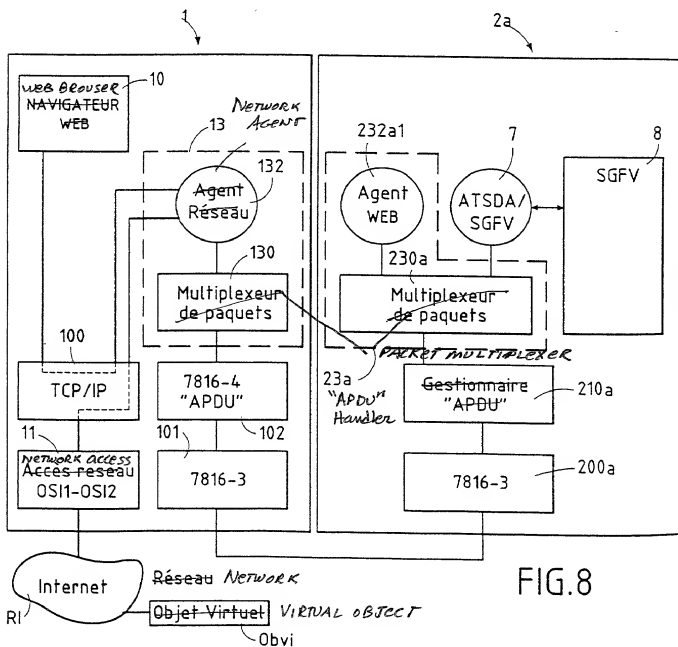


FIG. 7



678

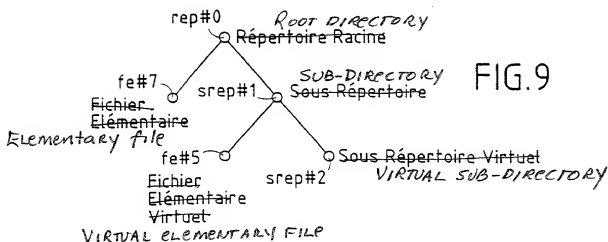


FIG. 9

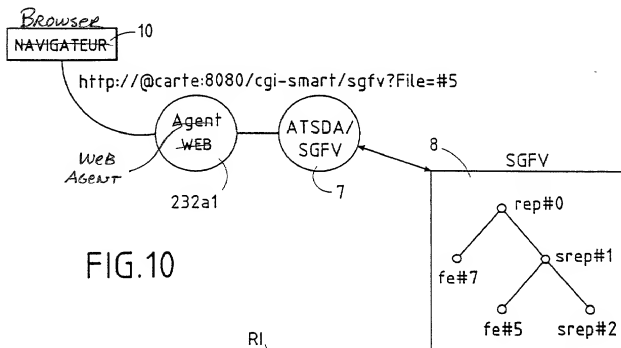


FIG. 10

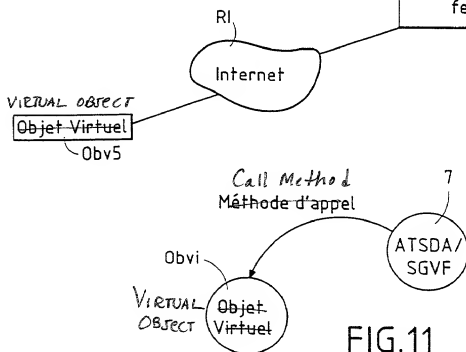


FIG. 11

7/8



LIST OF CALL STRUCTURES - Liste de structure d'appel  
 CALL STRUCTURE #1 - Structure d'appel #1  
 AGENT IDENTIFIER #1 - Identificateur agent #1  
 ARGUMENT #1 - Argument #1  
 CALL STRUCTURE #2 - Structure d'appel #2  
 AGENT IDENTIFIER #2 - Identificateur agent #2  
 ARGUMENT #2 - Argument #2

232am  
 LIST OF CALL STRUCTURE  
 CALL STRUCTURE #2  
 AGENT IDENTIFIER #2  
 ARGUMENT #2  
 Liste de structure d'appel  
 Structure d'appel #2  
 Identificateur agent #2  
 Argument #2

232an

Agent #2

FIG.12

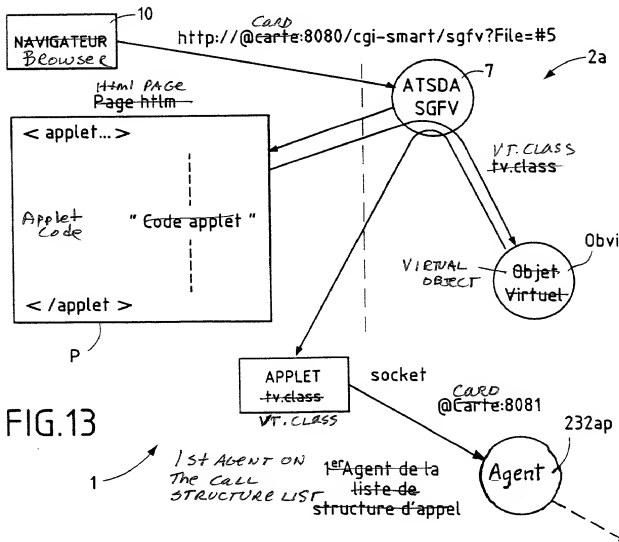


FIG.13

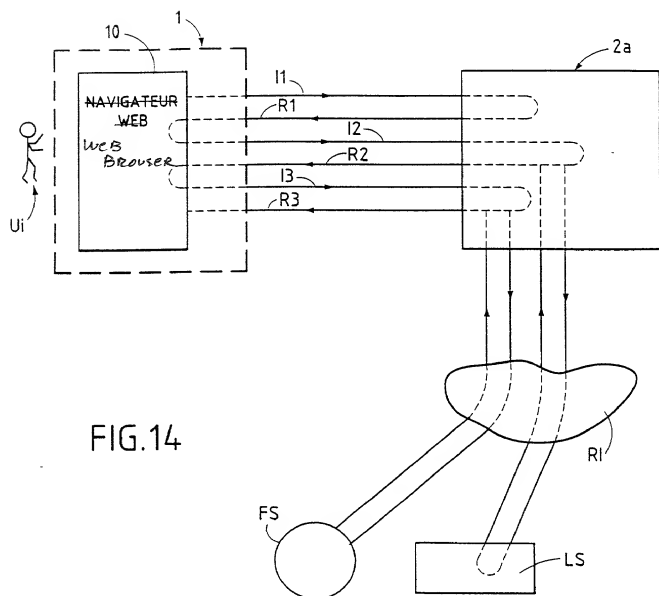


FIG.14

**METHOD FOR ACCESSING AN OBJECT USING A "WEB" BROWSER  
COOPERATING WITH A SMART CARD, AND  
ARCHITECTURE FOR IMPLEMENTING THE METHOD**

5 The invention relates to an embedded system containing information that makes it possible to instantiate an object located in a network, and a method for instantiating this object.

More specifically, the invention relates to a method for secure access to this object.

10 Within the scope of the invention, the term "object" should be considered in its most general sense. It includes many types of computer resources, such as text files, image files, or multimedia files (video, sound, etc.). It also includes transactions or connections to a computer system based on a given protocol.

15 In the first case, the objects will be considered herein to be static, since their instances are not time-dependent. In the second case, the objects will be said to be dynamic, since their instances vary with time. A non-limiting example, within the framework of an internet network, would be a "Telnet" connection.

20 Also within the scope of the invention, the term "user station" should also be understood in a general sense. The aforementioned user station may be constituted, in particular, by a personal computer running on various operating systems, such as WINDOWS or UNIX (both of which are registered trademarks). It can also be constituted by a workstation, a portable computer, or a so-called "dedicated" card terminal.

25 Likewise, within the scope of the invention, the term "network" includes any network comprising a set of servers linked to one another, particularly a global network in which information is transported end-to-end. It specifically includes the Internet, any network in which data is exchanged using an Internet protocol, private enterprise or similar networks known as "intranets," and the networks that extend them to the outside, known as "extranets." It could also be a GSM (Global System Mobile), ATM, UMTS, or GPRS (Global Packet Radio System) network, or a so-called "Wireless Network," for example 13E 802.11.

30 Hereinafter, without in any way limiting its scope, we will focus on the preferred application of the invention, unless otherwise indicated. We will therefore consider a user station, which will simply be called a "terminal," equipped with a smart card reader and connected to an internet network.

A chip-card based application system generally comprises the following main elements:

- a smart card;
- a host system constituting the aforementioned terminal;
- a communication network, i.e. the internet network in the preferred application;
- and an application server connected to the network.

Fig. 1A schematically illustrates an exemplary architecture of this type. The terminal 1, for example a personal computer, comprises a smart card 2 reader 3. This reader 3 may or may not be physically integrated into the terminal 1. The smart card 2 includes an integrated circuit 20 whose input-output connections are present on the surface of its substrate so as to allow an electric power supply and communications with the terminal 1. The latter comprises circuits for accessing a data transmission network *RI*. These circuits depend, in particular, on the nature of the network *RI* and the terminal 1. For example, they could comprise a network card for a local area network or a modem for connecting to a switched telephone line or to an Integrated Services Digital Network ("ISDN") for connecting to the Internet, for example via an Internet Service provider ("ISP").

The terminal 1 naturally comprises all of the circuits and components required for its proper operation, which have not been represented in order to simplify the drawing: central processor, RAM and ROM, magnetic disk mass storage, diskette drive and/or CD-ROM, etc.

Normally, the terminal 1 is also connected to standard peripherals, whether integrated or not, such as a display screen 5 and a keyboard 6.

The terminal 1 can be placed in communication with servers or any computer systems connected to the network *RI*, one of which 4 is illustrated in Fig. 1A. "Server" is understood to mean any information server capable of handling communication protocols, either to provide access to documents, or to provide access to machines. In the case of the preferred application of the invention, the access circuits 11 place the terminal 1 in communication with the servers 4 using a particular piece of software 10, called a "web browser." The latter makes it possible to access various applications distributed throughout the network *RI*, generally in a "client-server" mode. "Browser" is understood to indicate any means offering the following functions:

- the display of a page, particularly a page in "SGML" (Standard Generalized Markup Language);
- the downloading of resources offered on the page.



This function corresponds to what is meant by the term "browser." An SGML page contains presentation attributes, and links to other SGML documents, or "hyperlinks" to the outside world, i.e. URIs (Universal Resource Identifiers).

The SGML language is known to include several dialects, including HTML, XML, and WML.

Normally, communications in networks take place based on protocols that conform to standards comprising several superposed software layers. In the case of an internet network *RI*, the communications take place based on protocols specific to this type of communication, which will be described in detail below, and which also comprise several software layers. The communication protocol is chosen based on the specific application envisioned: interrogation of "web" pages, file transfers, electronic mail (or "e-mail"), forums or "news," etc.

The logical architecture of the system, which comprises a terminal, a smart card reader and the smart card, is represented schematically by Fig. 1B. It is described by the ISO 7816 standard, which itself comprises several sub-sections:

- ISO 7816-1 and 7816-2 related to the dimensions and the marking of the cards;
- ISO 7816-3 related to the transfer of data between the terminal and the smart card; and
- ISO 7816-4 related to the structure of the set of commands and the format of the commands.

In Fig. 1B, on the terminal 1 end, only the layers corresponding to the ISO 7816-3 standard, referenced 101, and the "APDU" command handler (ISO 7816-4 standard, referenced 201, are represented. On the smart card 2 end, the layers corresponding to the ISO 7816-3 standard are referenced 200 and the "APDU" command handler (ISO 7816-4 standard) is referenced 201. The applications are referenced  $A_1, \dots, A_i, \dots A_n$ ;  $n$  being the maximum number of applications present in the smart card 2.

A "cardlet" (registered trademark) application  $A_i$  present in the smart card 2 (Fig. 1A) dialogues with the terminal 1 using a set of commands. This set typically includes write commands and read commands. The format of the commands is known by the abbreviation "APDU" (for "Application Protocol Data Unit"). It is defined by the aforementioned ISO 7816-4 standard. A command "APDU" is written "*APDU.command*" and a response "APDU" is written "*APDU.response*". The "APDUs" are exchanged between the card reader and the smart card using a protocol specified by the aforementioned ISO 7816-3 standard (for example in character mode:  $T=0$ ; or in block mode:  $T=1$ ).

When the smart card 2 includes several distinct applications, as illustrated in Fig. 1B, it is said to be a multi-applicative card. However, the terminal 1 dialogues with only one application at a time. An application  $A_i$  is present, for example, in the form of a piece of software called an "applet," in "Java" (registered trademark) language, which will hereinafter be called a "cardlet." The selection of a particular "cardlet"  $A_i$  is obtained by means of an "APDU" of the selection type ("SELECT"). Once this choice has been made, the "APDUs" that follow it are routed to this "cardlet." A new "SELECT APDU" will have the effect of aborting the application in progress and choosing another one. The software subsystem that handles the "APDUs" 201 makes it possible to choose a particular application  $A_i$  in the smart card 2, to store the application thus chosen, and to transmit and/or receive "APDUs" to and from this application.

To summarize what has just been described, the selection of an application  $A_i$  and the dialogue with the latter are achieved through exchanges of "ADPU" commands. It is assumed that the applications  $A_i$  are conventional applications, which will hereinafter be called "GCAs" (for "Generic Card Applications").

In a chip-card based application system as illustrated by the architecture of Fig. 1B, various functions can be devolved to the smart card, especially security functions. It is actually advantageous to store the data linked to security (passwords, access rights, etc.) in a smart card that can be retained by the user. Furthermore, the data being stored in a read-only memory in a form that can be encrypted, it is not easily modifiable, or even directly readable from the outside.

However, it must be noted that the card 3 cannot communicate directly with the browsers on the market, unless the code of these browsers is modified. The current smart cards, which also conform to the standards mentioned above, have a hardware and software configuration that does not allow them to communicate directly with the Internet network either. In particular, they cannot receive and transmit data packets using any of the protocols used in this type of network. It is therefore necessary to provide an additional piece of software, installed in the terminal 1, generally in the form of what is called a "plug-in." This piece of software, which has the reference 12 in Fig. 1A, provides the interface between the browser 10 and the card 2, more precisely the electronic circuits 20 of this card 2.

In the current state of the art, the host system associated with the card reader 3, i.e. the terminal 1, is also associated with a particular application. In other words, it is necessary to provide a specific, so-called "dedicated" terminal for each specific application.

Furthermore, it is clear that, even given the rapid past evolution of the technologies and their foreseeable future evolution, the capacity for storing information in the random access or read-only storage circuits of a smart card remains, and will remain, very limited as compared to that offered by a terminal "hosting" this smart card, and naturally to those offered by larger systems, "minicomputers" or giant so-called "mainframe" systems. Also, it is not possible to store the data of a large number of applications in a smart card, particularly very large multimedia files.

The object of the invention is to eliminate the drawbacks of the devices of the prior art, some of which have been summarized, while meeting the needs that continue to arise. It must be possible, in particular, to be able to access a large number of applications, even those with a large quantity of data, of various natures and distributed throughout the Internet. Moreover, in a preferred embodiment, the accesses should benefit from maximum security, i.e. in practice, should take place via, and under the control of, a smart card containing all the data required to protect data exchanges. Finally, these accesses must be able to be achieved via a browser on the market and be transparent for a user, who should "see" the smart card as the only source, no matter where the application is stored.

According to a first characteristic of the method, the smart card presents the host system, i.e. the terminal, with a virtual terminal module, for example in the form of a page in "HTML" (HyperText Markup Language), or more generally in hypertext language, or even in the form of an "applet" in "Java" (registered trademark) language, which allows the user to choose a particular application from among those available and offered by the smart card. As a result, the terminal is generalized and supports a plurality of applications. The host system is seen as a peripheral of the smart card, and it makes hardware resources, such as a display screen, a keyboard, etc., available to the card.

To do this, a specific software communication layer is provided in the smart card and its counterpart in the terminal. The term "specific" should be understood to mean specific to the method of the invention. In essence, these so-called specific communication layers are generalized no matter what the application in question. They come into play only in the two-way data exchange process between the smart card and the terminal, and between the smart card and the network.

The specific software communication layers comprise, in particular, software components called "intelligent agents," which specifically allow protocol conversions. There are matching agents in the respective specific communication layers associated with the

terminal and with the smart card. According to the method of the invention, sessions are established between matching agents.

According to a second characteristic, the method of the invention makes it possible to activate conventional applications, i.e. of the aforementioned "CGA" type, located in a smart card, without having to modify them in any way.

To do this, one or more intelligent agents called script translators are provided, which receive requests from a browser and translate them into "APDU" commands comprehensible to the "CGA" application. This technical characteristic makes it possible to install, in a smart card whose architecture conforms to the method of the invention, a mechanism similar to the so-called "CGI" (Common Gateway Interface) function installed in conventional "web" servers.

Finally, according to another characteristic of the method of the invention, by implementing the aforementioned functions and mechanisms, it is possible to access computer resources distributed in a data transmission network to which the terminal is connected, particularly the Internet or a network of an equivalent type (intranet, extranet), without the user's having to know their locations. Hereinafter, as indicated, these resources will be called static or dynamic "virtual objects."

To do this, a script translating intelligent agent dedicated to this task is implemented, which cooperates with the other intelligent agents present in the terminal and/or the smart card. This agent makes it possible to define the virtual objects that the smart card, and thus the user (or holder of the smart card) can access, and provides the interrogating browser, via the smart card, with methods that make it possible to access these virtual objects.

The invention therefore relates to an embedded system, equipped with a chip comprising information processing means and information storage means and designed to cooperate with a network through a terminal, characterized in that:

- it stores at least one object file containing information associated with an object located in the network and making it possible to create an instance of this object;
- it comprises network interface means designed to cooperate with matching network interface means located in the terminal, so that the embedded system constitutes an information server in the network; and
- it comprises object file interface means, designed to establish a correspondence between information passing through the network interface means and assigned to at least said object file, and information exchanged with said object file.

Advantageously, the object file includes a piece of autonomous software executable in browser software. Advantageously, this piece of autonomous software is capable of implementing an object file management system of the embedded system.

Advantageously, said object file includes a description of actions to be performed in order to instantiate an object. Advantageously, the actions include actions performed inside the embedded system and consisting in sessions between agents of the embedded system.

Advantageously, the actions include actions performed outside the embedded system and consisting in sessions with agents of the terminal in order to obtain information from information servers of the network.

Advantageously, said network interface means are designed to cooperate with the matching network interface means located in the terminal, so that the embedded system acts like a client capable of connecting to at least one server of the network.

The invention also relates to a method for instantiating an object located in a network, using the aforementioned embedded system, characterized in that it makes it possible to describe a set of sessions between agents through an object file, by means of at least the following steps:

- establishing a list of the agents implemented;
- for each agent, defining call arguments necessary to the agent.

Advantageously, a call argument describes the opening of a session with another agent.

Advantageously, an agent modifies the list of arguments used by another agent.

In a variant, the method is characterized in that it implements sessions between agents described by an object file executed from the information server of the embedded system by means of at least the following steps:

- identification of an object file;
- execution of this object file.

Advantageously, the object file is identified by a particular directory name.

Advantageously, the object file is identified by a particular naming convention.

Advantageously, the object file is executed by instantiating the first agent associated with the object file.

Advantageously, the object file is executed by instantiating one or more agents referenced by the object file.

In a variant, the method is characterized in that it implements sessions between agents described by an object file executed from browser software by means of the following steps:

- loading by the browser software of an object file and a specific file capable of implementing it;

- execution of the specific software by the browser software.

Advantageously, the specific software is embodied in any interpreted language executable by the browser software.

Advantageously, the object file interpreter is embodied in browser software.

In a variant, the method is characterized in that it enables the embedded system to make it possible to implement sessions between agents described by an object file executed from browser software, and in that it comprises the step that consists of identifying, by means of a universal resource identifier, a specific software for implementing the browser software.

Advantageously, the universal resource identifier is integrated into a hypertext document.

Advantageously, said hypertext document is contained in the embedded system.

Advantageously, said hypertext document is contained in an information server of the network, remote from the embedded system.

Advantageously, said specific software is loaded by a method available in the browser software and deduced from the universal resource identifier.

The invention also relates to an embedded system, equipped with a chip comprising information processing means and information storage means and designed to cooperate with a network through a terminal, characterized in that it comprises network interface means designed to cooperate with matching network interface means located in the terminal, so that the embedded system constitutes an information server in the network and/or acts like a client capable of connecting to at least one server of the network. The invention also relates to a terminal designed to cooperate with a network and comprising information processing means, information storage means, and means for cooperating with an embedded system equipped with a chip comprising information processing means and information storage means, characterized in that it comprises network interface means designed to cooperate with matching network interface means located in the embedded system, so that the embedded system constitutes an information server in the network and/or acts like a client capable of connecting to at least one server of the network. Advantageously, the terminal dynamically acquires said network interface means from the network via a software loading mechanism.

This could be, in particular, a "plug-in" mechanism. If the terminal does not know the extension of the file containing said software, it searches in a server for the software associated with this extension, usually called a "helper."

Advantageously, said matching network interface means, in the terminal and in the embedded system, constitute a stack comprising one or more communication layers such that they allow the embedded system to share all or some of the communication layers of the terminal. Moreover, the terminal advantageously has access points in its communication layers, which allow it to route a flow of information to or from one or more of these layers. These access points correspond to the points known by the name "SAP" (Service Access Point) defined by the ISO standard.

The invention also relates to an embedded system, equipped with a chip comprising information processing means and information storage means and designed to cooperate with a network through a terminal, characterized in that it comprises network interface means designed to cooperate with matching network interface means located in the terminal, so that at least part of a flow of information exchanged between an application of the terminal and the network passes through the network interface means of the embedded system, in accordance with criteria known by the terminal. The invention also relates to a terminal designed to cooperate with a network and comprising information processing means, information storage means, and means for cooperating with an embedded system, equipped with a chip comprising information processing means and information storage means, characterized in that it comprises network interface means designed to cooperate with matching network interface means located in the embedded system, so that at least part of a flow of information exchanged between an application of the terminal and the network passes through the network interface means of the embedded system, in accordance with criteria known by the terminal. The routing of part of the flow of information to the embedded system is advantageously performed by the information processing means of the terminal in accordance with pre-established criteria, either statically or in a way that is negotiated by a dialog with an embedded system; in the latter case, the terminal can, for example, ask the embedded system for its "IP" address (if it is the Internet), using known protocols.

Advantageously, the aforementioned criteria include one of the following:

- the "IP" address of the embedded system, or its "ATM" address in the case of an ATM network;

- the IP address of the terminal and a particular "TCP" or "UDP" port;

- any SAP access point that references the embedded system.

The invention will now be described in greater detail in reference to the attached drawings, in which:

- Figs. 1A and 1B schematically illustrate the hardware and software architectures, respectively, of an exemplary chip-card based application system according to the prior art;

- Fig. 2 schematically illustrates an exemplary chip-card based application system according to the invention, the latter acting as a "web" server;

- Fig. 3 illustrates, in simplified fashion, the logical architecture of a system in which the smart card comprises intelligent agents;

- Fig. 4 illustrates a system architecture according to the invention, in which the smart card comprises script translating intelligent agents;

- Fig. 5 is a diagram schematically illustrating the main phases of exchange between a browser and a smart card having the architecture of Fig. 4;

- Fig. 6 is a diagram schematically illustrating an essential aspect of the method according to the invention through which it is possible to access virtual objects distributed in an internet network via a smart card and a "web" browser;

- Fig. 7 schematically illustrates the organization of a so-called virtual file management system for implementing this aspect of the method of the invention;

- Fig. 8 is an exemplary architecture comprising a virtual file management system according to Fig. 7;

- Figs. 9 through 15 are diagrams schematically illustrating several embodiments of the method according to the invention.

Before describing the method for activating applications located in a smart card according to the invention and detailing an architecture for its implementation, it is first appropriate to briefly summarize the chief characteristics of communication protocols in networks.

The architecture of communication networks is described in various layers. For example, the "OSI" ("Open Systems Interconnection") standard defined by the "ISO" comprises seven layers, which run from the so-called lower layers (for example the so-called "physical" layer that supports physical transmission) to the so-called upper layers (for example the so-called "application" layer), passing through intermediate layers, including the so-called "transport" layer. A data layer offers its services to the layer that is immediately above it and requires other services from the layer immediately below it, via appropriate



interfaces. The layers communicate using primitives. They can also communicate with layers of the same level. In certain architectures, some of these layers may be non-existent.

In an Internet type environment, there are five layers, and more precisely, from the top layer to the bottom layer: the application layer ("http", "ftp", "e-mail", etc.), the transport layer ("TCP"), the network address layer ("IP"), the data link layer ("PPP", "SLIP", etc.) and the physical layer.

Having given this summary, we will now describe an architecture of a smart card-based application system that enables the smart card to act as a "web" server. An example of such an architecture is represented schematically in Fig. 2. The elements common to Figs. 1A and 1B have the same references and will be re-described only as necessary. In order to simplify the drawing, the various peripherals connected to the terminal (Fig. 1A: screen 5 and keyboard 6, for example) are not represented.

With the exception of specific software communication protocol layers referenced 13 and 23a, respectively installed in the terminal 1 and the smart card 2a, the other hardware and software elements are common to the prior art.

The terminal 1 comprises circuits 11 for accessing the network *RI*, constituted for example by a modem for the Internet or a network card for a local area network. These circuits contain the lower software layers *C*<sub>1</sub> and *C*<sub>2</sub> corresponding to the "physical" and "data link" layers.

Also represented are the upper layers *C*<sub>3</sub> and *C*<sub>4</sub>, corresponding to the "network address" ("IP" in the case of the Internet) and "transport" ("TCP") layers. The top application layer ("http", "ftp", "e-mail", etc.) is not represented.

The interface between the lower layers *C*<sub>1</sub> and *C*<sub>2</sub> and the upper layers *C*<sub>3</sub> and *C*<sub>4</sub> is constituted by a software layer generally called a "lower level driver." The upper layers *C*<sub>3</sub> and *C*<sub>4</sub> rely on this interface and are implemented by means of specific function libraries or network libraries 14, to which they correspond. In the case of the Internet, "TCP/IP" is implemented by means of libraries known as "sockets."

This organization enables a browser 10 (Fig. 1A) to submit requests to a server 4 (Fig. 1A) in order to consult "web" pages ("HTTP" protocol), transferring files ("FTP" protocol) or sending e-mail ("e-mail" protocol), in an entirely conventional way.

The terminal 1 also comprises a card reader 3, which may or may not be integrated. In order to communicate with the smart card 2a, the card reader also includes two lower layers *CC*<sub>1</sub> (physical layer) and *CC*<sub>2</sub> (data link layer), which play a role similar to the layers *C*<sub>1</sub> and

C<sub>2</sub> The software interfaces with the layers CC<sub>1</sub> and CC<sub>2</sub> are described, for example by the "PC/SC" specification ("part 6, Service Provider"). The layers CC<sub>1</sub> and CC<sub>2</sub> themselves are described by the ISO 7816-1 through 7816-4 standards, as has been indicated.

5 An additional software layer 16 forms an interface between the applicative layers (not represented) and the lower layers CC<sub>1</sub> and CC<sub>2</sub>. The main function devolved to this layer is a multiplexing/demultiplexing function.

The communications with the smart card 2a take place according to a paradigm similar to that used to handle files in an operating system of the "UNIX" (registered trademark) type: "OPEN", "READ", "WRITE", "CLOSE", etc.

10 On the smart card end 2a, there is a similar organization, i.e. the presence of two lower layers, referenced CCA<sub>1</sub> (physical layer) and CCA<sub>2</sub> (data link layer), as well as an interface layer 26a, entirely similar to the layer 16.

According to a first characteristic, on both ends, i.e. in the terminal 1 and in the smart card 2a, two specific protocol layers are provided, respectively 13 and 23a.

15 In the terminal 1, the specific layer 13 interfaces with the "lower level drivers" 15, with the libraries 14 of the network layers C<sub>3</sub> and C<sub>4</sub>, and with the protocol layers of the card reader 3, i.e. the lower layers CC<sub>1</sub> and CC<sub>2</sub>, via the multiplexing layer 16. The specific layer 13 allows the transfer of network packets to and from the smart card 2a. In addition, it adapts existing applications such as the Internet browser 10 (Fig. 2), the e-mail software, etc., for  
20 utilizations involving the smart card 2a.

On the smart card end 2a, there is an entirely similar organization constituted by an additional instance of the specific layer, referenced 23a, the counterpart of the layer 13.

More precisely, the specific layers 13 and 23a are subdivided into three main software elements:

- 25 - a module 130 or 230a for transferring blocks of information between the layers 13 and 23a, via the conventional layers CC<sub>1</sub>, CC<sub>2</sub>, CCA<sub>1</sub> and CCA<sub>2</sub>;
- one or more pieces of software called "intelligent agents," 132 or 232a, which perform, for example, protocol conversion functions;
- and a module for managing the specific configuration 131 and 231a, respectively,  
30 which module may be integrated into a particular intelligent agent.

Hence, in the terminal 1 and the smart card 2a, there is a communication protocol stack between the two entities.

The level-2 layers (data link layers)  $CC_2$  and  $CCa_2$  handle the exchange between the smart card 2a and the terminal 1. These layers are responsible for the detection and possible correction of transmission errors. Various protocols are usable, the following being a non-exhaustive list of examples:

- the ETSI GSM 11.11 recommendation;
- the protocol defined by the ISO 7816-3 standard, in character mode  $T=0$ ;
- the protocol defined by the ISO 7816-3 standard, in block mode  $T=1$ ;
- or the protocol defined by the ISO 3309 standard, in "HDLC" (for "High-Level Data Link Control") frame mode.

For purposes of the invention, it is preferable to use the ISO 7816-3 protocol, in block mode.

In an essentially known way, each protocol layer is associated with a certain number of primitives that allow the data exchanges between layers of the same level and from one layer to another. For example, the primitives associated with the level-2 layer are of the "data request" ("Data.request"), "data response" by the card ("Data.response"), and "data confirm" ("Data.confirm") types, etc.

More specifically, the layers 13 and 23a are responsible for the dialogue between the smart card 2a and the host, i.e. the terminal 1. These layers allow the exchange of information between a user (not represented) of the terminal 1 and the smart card 2a, for example via drop-down menus in the form of hypertext in the "HTML" format. 3. They also allow the implementation of a configuration adapted for the sending and/or receiving of data packets.

As indicated above, the layers comprise three distinct entities.

The first layer, 130 or 230a, is essentially constituted by a software multiplexer. It allows the exchange of information between the smart card 2a and the host terminal 1, in the form of protocol data units. It plays a role similar to that of a data packet switcher. These units are sent or received via the level-2 layer (data link layer). This particular communication protocol makes it possible to put at least one pair of "intelligent agents" in communication with each other. The first agent of each pair, 132, is located in the layer 13, on the terminal 1 end, the second 232a, is located in the layer 23i on the smart card 2a end. A link between two "intelligent agents" is associated with a session. A session is a two-way data exchange between these two agents.

An intelligent agent can perform all or some of the functions of the level-3 and 4 layers, depending on the configuration used by the terminal 1.

A particular intelligent agent is advantageously identified by a whole number, for example in 16 bits (a number between 0 and 65535). This identifier is used, for example, in a protocol data unit constituting a destination reference and a source reference.

There are two main categories of intelligent agents: agents of the "server" type, which are identified by a fixed reference, and agents of the "client" type, which are identified by a variable reference delivered by the configuration management module 131 or 231a.

The process for opening a session is normally the following: an intelligent agent of the "client" type opens the session with an intelligent agent of the "server" type. The layers 130 and 230a manage tables (not represented) that contain a list of the intelligent agents present on the host terminal 1 end and on the smart card 2a end.

The intelligent agents are associated with specific properties or attributes. To illustrate the concept, and to give a non-limiting example, the following six properties are associated with the intelligent agents:

- "host": agent located in the terminal;
- "card": agent located in the smart card;
- "local": agent not communicating with the network;
- "network": agent communicating with the network;
- "client": agent that initiates a session;
- "server": agent that receives a session request.

The intelligent agents make it possible to exchange data (hypertext, for example), but also to initiate network transactions.

The configuration management modules, 131 and 231a, respectively, can be integrated, as has been indicated, into specific intelligent agents. For example, the module 131 on the host terminal 1 end, specifically manages information related to the configuration of this terminal (operating modes), the lists of other agents present, etc. The module 231a on the smart card 2a end has similar functions. These two intelligent agents can be placed in communication with one another in order to establish a session.

According to one characteristic, the smart card 2a offers the host system, i.e., the terminal 1, a virtual terminal model. To do this, the smart card 2a acts as a "web" server.

The smart card 2a is "addressed" by the browser 10. It then transfers to the browser a "web" page in "HTML" language, an "applet" or any other piece of software. For example, the "web" page can be presented in the form of a welcome page that gives a choice of possible applications and/or hyperlinks to external servers.

In a practical way, the smart card 2a is advantageously "addressed" using a "URL" (for "Universal Resource Locator") address defining a loopback to the terminal 1 itself, and not pointing to an external server. For example, the structure of this "URL" is normally as follows:

http://127.0.0.1:8080 (1)

in which 127.0.0.1 is the "IP" loopback address and 8080 is the port number.

Fig. 3 illustrates, in simplified fashion, the logical architecture of a system wherein the smart card 2a comprises intelligent agents, only two of which are represented: an intelligent agent of a type not precisely defined 232a<sub>2</sub> and an intelligent agent 232a<sub>1</sub> of the "web" type. The logical stack comprises the lower protocol layers, referenced 200a, which comply with the ISO 7816-3 standard (Fig. 2: CCA<sub>1</sub> and CCA<sub>2</sub>), the "APDU" command handler 201a<sub>1</sub>, and the packet multiplexer 230a, the latter being interfaced with the intelligent agents, particularly the "web" intelligent agent 232a<sub>1</sub>.

On the terminal end, there are two stacks, one communicating with the internet network RI, the other with the smart card 2a. The first stack comprises the elements 11 (Fig. 2: C<sub>1</sub> and C<sub>2</sub>) for accessing the network (OSI 1 and 2 standards) and the "TCP/IP" protocol layers (Fig. 2: C<sub>3</sub> and C<sub>4</sub>), referenced 100. The latter layers are interfaced with the "web" browser 10. The other stack comprises the lower protocol layers, referenced 101, which comply with the ISO 7816-3 standard (Fig. 2: C<sub>1</sub> and C<sub>2</sub>), the "APDU" command manager 102 and the packet multiplexer 130, the latter being interfaced with intelligent agents, only one of which 132 is represented. The latter, which is assumed to be of the "network" type, can also communicate with the browser 10 via the "TCP/IP" layers 100, and with the internet network RI via these same "TCP/IP" layers 100 and the element 11 for accessing the network RI.

The "APDU" command handler 201a is also interfaced with one or more layers on the applications level, which will simply be called applications. These applications, as indicated, are applications of the conventional, so-called "cardlet" type.

In summary, the "web server" function provided by the smart card 2a can be produced by associating the "web" intelligent agent 232a<sub>1</sub> in the smart card with the network agent 132 in the terminal.

The smart card 2a then actually has the "web" server functionality. Moreover, according to one characteristic of the method of the invention, any conventional application A<sub>1</sub> through A<sub>n</sub>, of the aforementioned "CGA" type, can be activated through this "web"

server, either through the "web" browser 10 present in the terminal 1, or through a remote browser located at any point in the internet network *RI*. According to the method of the invention, the applications  $A_1$  through  $A_n$  do not need to be rewritten and are used as they are.

According to another characteristic of the invention, these applications remain  
5 accessible to a terminal of the conventional type, i.e. according to the prior art.

In order to meet these requirements, the "web" server function offered by the smart card 2a includes a mechanism similar to the so-called "CGI" ("Common Gateway Interface") function installed in conventional "web" servers.

Before describing an exemplary architecture according to the invention that makes it  
10 possible to produce a function of this type in the smart card itself, it is appropriate to review the chief characteristics of a "CGI" operating mode.

"CGI" is a specification for implementing, from a "web" server, applications written for the "UNIX" (registered trademark), "DOS", or "WINDOWS" (registered trademark) operating systems. For example, for the "UNIX" operating system, the specification is "CGI 1.1" and for the "WINDOWS 95" operating system, the specification is "CGI 1.3".  
15

Again by way of example, an "HTTP" request to a "URL" address such as:

`http://www.host.com/cgi-bin/xxx.cgi` (2),

in which "host" refers to a host system (generally remote), is interpreted by a "web" server as the execution of a command script of the "CGI" type named "xxx" and present in the  
20 directory "cgi-bin" of this host system. Although the name of the directory could theoretically be any name, conventionally it is the name given to the directory storing the "CGI" type scripts. A script is an instruction sequence of the operating system of the host system whose final result is transmitted to the "web" browser that sent the aforementioned request. Various languages can be used to write this script, for example the "PERL"  
25 (registered trademark) language.

In a practical way, the request is normally displayed on a computer screen in the form of a form included in an "HTML" page. The "HTML" language makes it possible to translate a form into a "URL" address. The form includes one or more fields, which may or may not be required, that are filled in by a user using the customary entry means: a keyboard for text,  
30 a mouse for boxes to be checked or so-called "radio" buttons, etc. The content of the form (possibly along with so-called "hidden" information and instructions) is sent to the address of the "web" server. The "HTML" code of the page describes the physical structure of the form

(frame, graphics, color, and any other attribute) as well as the structure of the fields of data to be entered (name, length, data type etc.).

The transmission can take place based on two main types of formats. A first format uses the so-called "POST" method and a second uses the so-called "GET" method. Format type information is present in the code of the form page.

This mechanism, however, is not directly transposable to a smart card, even when the latter offers the "web" server functionality according to one of the characteristics of the invention.

We will now describe an exemplary architecture that makes it possible to activate any conventional type of application via a "web" server in the smart card 2a, in reference to Fig. 4.

In a first step, a user (not represented) calls, from his "web" browser (Fig. 3: 10), a "URL" address that may be presented in the following way:

`http://@card:8080/xxx.html` (3),

in which "@card" is an IP address of the smart card (for example the loopback address "127.0.01" described above: see formula (1)), and "xxx.html" is a page in "HTML" language related to a particular application "xxx" offered by the smart card.

In a second step, in the manner described above, the smart card returns an "HTML" page, for example of the form type.

In a third step, the user fills in the fields of the form and transmits its contents to the smart card, usually by clicking on a particular field of the "push button" type.

The data is then sent and received by the network agent 132. The data then passes through the packet multiplexer 130 (which constitutes one of the components of the specific layer 13 on the terminal 1 end), the "APDU" command handler 102, the protocol layers 101, in order to be transmitted to the smart card 2a. It then passes through the protocol layers 200a, the "SPDU" command handler 201a, the packet multiplexer 230a in order to be received by the "web" agent 232a<sub>1</sub>. Thus, a logical session is established between the two intelligent agents, as explained above.

It is appropriate to note that the data addressed to the "web" agent 232a<sub>1</sub> is transported, in an essentially conventional way, in the form of "APDU" commands addressed to the particular "Packet Multiplexer" application. The "APDU" command handler 201a selects this application in a way entirely similar to the other applications of the "CGA" type

present in the smart card 2a, referenced  $A_1$  through  $A_n$ . In other words, the packet multiplexer 230a is seen by the "APDU" command handler 201a as an ordinary "CGA" application.

The "HTTP" request is then analyzed by the "web" agent 232a<sub>1</sub>, which detects a reference to a particular directory, which will hereinafter be called, conventionally, "cgi-smart", and to a particular application, for example "xxx" in the case of the example described. The complete path in this case is therefore "cgi-smart/xxx".

According to one characteristic of the method of the invention, the above entity designates a particular script associated with an equally particular application "xxx".

In a fourth step, the script is then interpreted by an intelligent agent called a "script translating agent," which will hereinafter be called "ATS". This translation can be performed in various ways:

- a/ by the "web" agent 232a<sub>1</sub> itself, which in this case is equipped with a double capacity;
- b/ by a unique script translating agent capable of translating all of the scripts present in the smart card 2a;
- c/ by a dedicated script agent that will hereinafter be called "ATSD" (one per script); or
- d/ by an "APDU" agent 2010a of the "APDU" command handler 201a, which in this case is equipped with a double capacity.

The "APDU" agent 2010a is a component of the "APDU" command handler layer 201a. The latter, as has been indicated, is a layer capable of centralizing all of the "APDU" commands sent and/or received by the system, of selecting applications from among  $A_1$  through  $A_n$ , but also of offering an interface of the intelligent agent type. It is therefore capable, according to one of the characteristics of the method, of communicating with all of the intelligent agents of the system (via sessions), whether these agents are located in the terminal 1 or the smart card 2a.

In case c/ above, a session is opened between the "web" agent 232a<sub>1</sub> and one of the "ATSD" agents.

Fig. 4 illustrates an exemplary architecture for which the translating agents are of the "ATSD" type. They are referenced  $ATS_1$  through  $ATS_n$  and associated with the applications  $A_1$  through  $A_n$ . Assuming that the application selected is the application  $A_i$ , the session is established between the "web" agent 232a<sub>1</sub> and the agent  $ATS_i$ .



A script translating agent generates a set of "APDU" commands. A session is opened between the translating agent, for example the agent  $ATS_i$ , and the "APDU" agent 2010a. The commands are then sent to the "APDU" agent 2010a. The "APDU" command handler 201a selects the "CGA" application  $A_i$  (for example the "e-purse" application) and transmits it the "APDU" commands, commands that are translated and therefore conventional, which it is capable of understanding. This application is therefore correctly activated, without having to be modified or rewritten.

The responses from the "CGA" application  $A_i$  are transmitted to the "APDU" command handler 201a, to the "APDU" agent 2010a, then again to the agent  $ATS_i$  (and more generally to the script translating agent).

Based on the success or failure of the running of the script, the script translating agent, for example the agent  $ATS_i$  in the example of Fig. 4, generates a page in "HTML" language and transmits it via the various layers used by the initial request, but in the opposite direction, in order for it to be presented on the display screen 5 (Fig. 1A).

The various paths are represented symbolically in Fig. 4 by solid lines connecting the functional blocks or by dotted lines inside these blocks.

Fig. 5 schematically summarizes the main steps of the process that has just been described:

- a/ the transmission via the Internet network  $RI$  (or from the local terminal, in either case by means of a conventional browser 10), of an "HTTP" request, referenced  $RQ$ ;
- b/ a response from the "web" server of the smart card 2a, in the form of a form, referenced  $FO$ ;
- c/ the transmission of the filled-in form, in the form of a new request  $RQ$ ; and
- d/ a response in the form of an "HTML" page, referenced  $PR$ .

The response could also consist in the transmission of a file, or of a piece of software or "Applet".

By implementing the mechanisms and functions that have just been described, particularly the "web" server function and the use of intelligent script translating agents, according to an essential characteristic, the method according to the invention will make it possible to define a virtual environment, advantageously protected by the smart card. In a preferred embodiment, this environment is compatible with applications of the so-called multimedia type.

This last characteristic is particularly advantageous because the recent "web" browsers, which are entirely conventional, by their very nature make it possible to build multimedia environments (animated images, sounds, etc.). They are in fact associated with software tools, which may or may not be integrated, that make it possible to manipulate multimedia files (viewers, etc.). In any case, the browsers make it possible to download multimedia data files, which are usually large, and store them on a hard disk, for example in the terminal, or on a similar mass storage device. In particular, technologies have been proposed for displaying video sequences in real- or near-real time, or reproducing sound, from "web" sites on the Internet.

However, as has been noted, a smart card has only a small storage capacity. Moreover, it allows only a very low throughput of data during exchanges. It is therefore impossible to store a large number of data files in it. It is also practically impossible to store multimedia files, except for very short sequences or sound sequences encoded in a particular format, such as "MIDI" encoding.

Beyond these limitations of a technological nature, it is also desirable to be able to access remote applications, while enjoying a high level of security, which only the use of a smart card can offer.

The method according to the invention allows this mode of operation. The chip-card protected virtual multimedia environment, according to a preferred embodiment, makes it possible to:

- define virtual objects that the smart card can access;
- provide methods for accessing these objects.

Fig. 6 schematically illustrates this essential aspect of the method according to the invention.

A user  $U_i$  interrogates the smart card 2a using the "web" browser 10 contained in the terminal 1. Using a mechanism that will be described below, particularly by means of the "web server" function described above, the smart card 2a will return to the browser a list of so-called virtual objects  $Obv_i$ ,  $i$  being an arbitrary subscript, to which it has access, i.e. in practice, to which the smart card 2a or the user  $U_i$  has access rights. In essence, these access rights can be strictly linked to the smart card 2a and unchangeable. They can also be linked to a user profile, the user  $U_i$  supplying, for example, identification data and a password. The smart card 2a performs a verification through a comparison with data in a security database stored in a read-only memory, and if the result of the comparison is positive, supplies a list of

virtual objects  $Obv_i$  associated with the "identification data/password" pair. In an essentially known way, this first phase can implement a method for encrypting data exchanged between the terminal and the smart card 2a or implement a secure transmission protocol "HTTPS." The smart card 2a will also supply a list of methods for accessing the virtual objects  $Obv_i$ .

5       The virtual objects  $Obv_i$  which are either static or dynamic as indicated above, can be located either in the smart card 2a or in the terminal 1, or more generally in any system connected to the internet network  $RI$ . According to one characteristic of the invention, this location is "transparent" for the browser 10, and hence for the user  $U_n$ , as will be shown.

10       The method according to the invention specifically uses what will hereinafter be called a virtual file management system, or "SGFV," and a specialized script translating intelligent agent that will be called "ATSDA/SGFV," dedicated to this task. This intelligent agent supplies the list of virtual objects  $Obv_i$  that the smart card 2a can access. A particular "URL" address is associated with each virtual object  $Obv_i$ . The call-up of this "URL" from the "web" browser 10 makes it possible to instantiate the virtual object  $Obv_i$  using a given call  
15       method, which may or may not be specific to this object.

First, we will briefly summarize the chief characteristics of a conventional file management system, hereinafter called "SGF." Such a system is used to store information on a medium such as a hard disk. The information is stored in the form of a file. A file, whether pure data or program instructions, is conventionally composed of a set of fixed-size blocks. A  
20       well known mechanism makes it possible to obtain a list of the storage blocks that constitute the file and their addresses in the memory.

A directory is a particular file whose content is a list of file descriptors. Such a descriptor comprises, for example, the following elements:

- the name of the file;
- 25       - the length of the file;
- the creation date;
- a reference that makes it possible to retrieve a list of the blocks of the file (number of the first block, table of the block numbers, etc.); and
- 30       - attributes that specify particular properties of the file (directory, read, write, execution, etc.).

The first directory is normally called the root directory. A directory that is not a root is called a sub-directory. The directory that contains the descriptor of a given file is its father directory. The address of a file in the "SGF" is therefore a sequence of directory names, from

the root directory to the father directory of the file, which defines a path. For example, such a path appears as follows:

"root/directory1/directory2/file\_name" (4),

the numbers 1 and 2 being arbitrary, "root" being the name of the root directory, and "file\_name" being any file name.

For a smart card, the ISO 7816-4 standard defines the root directory called "MF" (for "Master File"), sub-directories called "DF" (for "Dedicated Files") and elementary files called "EF" (for "Elementary Files").

Within the scope of the invention, the file management system "SGFV", which will be referred to as "virtual," makes it possible to define virtual objects *Obv<sub>i</sub>*, to which the smart card 2a can have access. According to the method of the invention, a virtual object *Obv<sub>i</sub>* is associated with a virtual elementary file. The content of a virtual elementary file is constituted by the set of information that makes it possible to access the associated virtual object *Obv<sub>i</sub>* and to obtain an instance of same in the terminal 1.

In a practical way, as illustrated schematically by Fig. 7, the system "SGFV" can constitute a subset of a standard "SGF" system, and more precisely, an "SGFV" is contained in an elementary file, as defined by the aforementioned ISO 7816-4 standard.

A file descriptor generally comprises the following elements;

- the name of the file;
- the length of the file;
- the creation date;
- a reference (advantageously a whole number) that makes it possible to retrieve a list of the blocks of the file (number of the first block, table of block numbers, etc.): a virtual file is identified by its name or this unique reference; and
- attributes of the file that specify the particular references of the file: directory or elementary file, virtual or non-virtual, direct or indirect.

A "direct virtual object" is an object that is instantiated from the smart card 2a. It is typically a static virtual object *Obv<sub>i</sub>* that can be manipulated by the browser, for example displayed (image, etc.). An "indirect virtual object" is a virtual object *Obv<sub>i</sub>* that is instantiated from the browser 10, typically by means of an "applet."

Fig. 8 schematically illustrates the architecture of a chip-card system that makes it possible to instantiate a virtual object *Obv<sub>i</sub>* located anywhere in the internet network *RI* via

the browser 10 and the smart card 2a. The elements common to the previous figures have the same references, and will be re-described only as necessary.

The architecture illustrated in Fig. 8 is very similar to that of Fig. 4. The essential difference lies in the fact that it provides an "SGFV" 8 stored in the smart card 2a, and a specific script translating intelligent agent "ATSDA/SGFV", referenced 7. The operating mode is similar to that illustrated by Fig. 4 when wishing to access a particular application  $A_i$ . It is therefore unnecessary to re-describe it in detail. In the present case, the particular application is replaced by the virtual file management system "SGFV" 8. To begin with, a session is established between the network intelligent agent 132 and the "web" intelligent agent 232a<sub>1</sub>. According to the mechanism described above, a session is then established between the "web" agent 232a<sub>1</sub> and the intelligent agent "ATSDA/SGFV" 7.

In a practical way, the intelligent agent "ATSDA/SGFV" 7 is accessible using "URLs," typically of the following type:

"http://www.host.com/cgi-smart/sgfv?" (5)

in which "sgfv" is an application of the "CGI" type associated with the intelligent agent "ATSDA/SGFV" 7. The above request makes it possible to scan the tree of directories and "show" their content to the browser 10, by means of an "HTML" page. The "leaves" of the tree are virtual or non-virtual elementary files associated with a hyperlink. The transmission in the "smart card 2a-terminal 1" direction is performed as explained in connection with Fig. 4.

In other words, the intelligent agent "ATSDA/SGFV" 7 associates a "URL" address with any element of the "SGFV" 8, be it a directory or an elementary file. The "URL" address of a directory designates an "HTML" page that contains the list of its elements. The "URL" address of an elementary file makes it possible to create an instance of the virtual object  $Obv_i$  associated with this virtual file.

To illustrate the concept, if one uses the above "URL" address (5), one obtains an "HTML" page that presents the contents of the root directory to the browser 10. This root directory is constituted by a set of sub-directories and files, as illustrated schematically by Fig. 9. In this figure, we have represented a root directory  $rep\#0$ , at the top level, a real elementary file  $fe\#7$ , and a real sub-directory  $srep\#1$  at the lower intermediate level, and a virtual sub-directory  $rep\#2$  and a virtual elementary file  $fe\#5$  at the lowest level, both of which are dependent on the real sub-directory  $srep\#1$ , the reference numbers being purely arbitrary.

During a first phase, the intelligent agent "ATSDA/SGFV" 7 transmits to the browser 10, in response to the request received, an "HTML" page (not represented) showing, in one form or another, the hierarchical structure of the "SGFV" 8. The page is normally displayed on a display screen (Fig. 1A: 5), for example in the form of a menu. Each line of the menu is constituted by a hyperlink describing a sub-directory or an elementary file. The display can advantageously be in graphical form, and may or many not be associated with a descriptive text, the image of the tree of Fig. 9 being displayed on the aforementioned screen. It is also possible to display icons or complex (for example three-dimensional) forms, each being associated with one of the virtual objects to be instantiated, and possibly representing their nature (for example, a camera representing a video file), which may or may not be associated with a descriptive text.

The user  $U_i$  is prompted to click on a hyperlink (on a node or on a branch in the case of a graphical image). Through this action, he will be able to obtain an instance of the desired virtual object  $Obv_i$ .

The "SGFV" 8 is advantageously stored in a re-programmable memory contained in the smart card 2a, for example an "EEPROM" (electrically erasable memory), as illustrated schematically in Fig. 10. The "SGFV" 8 reproduces the structure of the tree of Fig. 9.

Again in the example described, having obtained the menu page, obtained during an initial phase by clicking on a "URL," which could typically be the following:

"http://www.host.com/cgi-smart/sgfv?/file#5 (6),

the user  $U_i$  obtains an instance of the virtual object  $Obv_5$  associated with the elementary file referenced  $fe\#5$  in Fig. 10. Similarly, he could have obtained the contents of a sub-directory, the parameter "file#5" being replaced with "file#x" in (6), #x being the number associated with the sub-directory.

The non-virtual files are stored in the smart card 2a and conform to the usual paradigm that governs "SGFs." They contain data, for example keys, necessary to the intelligent agent "ATSDA/SGFV" 7.

There are various possible conventions for defining the information required for the instantiation of a virtual object  $Obv_i$ , for example:

- a virtual file of null length inherits access methods from its father directory;
- a virtual directory is associated with a virtual elementary file whose name is keyed (for example "virtual"), and which contains the access methods of this directory.

In essence, in addition to the list of the accessible virtual objects  $Obv_i$ , an intelligent agent "ATSDA/SGFV" 7 must also supply a method for accessing a given virtual object, from all or part of the information contained in a virtual elementary file. Fig. 11 schematically illustrates this process.

According to the method of the invention, two access methods, respectively called direct and indirect, are provided, in accordance with the attributes of the virtual elementary file in question.

The direct method consists in a description of a chain of intelligent agents used in the process for accessing a virtual object  $Obv_i$  and for obtaining an instance of same in the terminal. When a session is opened, a given intelligent agent receives, from the agent that initiated this session, a list of call structures that will hereinafter be called a "call method" or "Method PDU" (for "Method Protocol Data Unit").

A call structure comprises:

- an identifier of the intelligent agent with which the session is opened;
- data, or arguments, required for its utilization.

The first intelligent agent addressed by the aforementioned list "consumes" a first call structure that is addressed to it. It transmits the rest of the structure list to the next intelligent agent, with which it establishes a session, until the end of the list is reached.

To illustrate the concept, an example of the various stages of exchange between the intelligent agent "ATSDA/SGFV" 7 and two cascaded intelligent agents,  $232a_m$  and  $232a_n$ , is illustrated schematically by Fig. 12. The call structure list sent by the intelligent agent "ATSDA/SGFV" 7 actually comprises two distinct sub-lists, respectively referenced #1 and #2 in their headers. The first one is consumed by the first intelligent agent  $232a_m$ , and the second by the second intelligent agent  $232a_n$ . An intelligent agent, for example the intelligent agent  $232a_m$ , is identified by a reference, or agent identifier ("Agent Identifier #1" or "Agent Identifier #2"). The intelligent agent addressed, with which a session is established, retains the sub-list that is addressed to it by means of the header ("Call Structure #1" or "Call Structure #2"). The arguments of the sub-list that it retains ("Argument #1" or "Argument #2") are constituted by a set of data necessary to the proper functioning of this agent. For example, a piece of data could be a (non-virtual or virtual direct) file name.

A given intelligent agent, for example the intelligent agent  $232a_m$ , can modify the rest of the call structure list before transmitting it to the next intelligent agent,  $232a_n$ . To do this, it addresses this intelligent agent  $232a_n$  and establishes a session with it.

The call method can advantageously be described using the the ASN.1 language (the ISO's "Abstract Syntax Notation 1").

The direct access method makes it possible to definitively instantiate a virtual object *Obv<sub>i</sub>* directly from the smart card 2*a*. *A priori*, it is a static object. The instantiated object is normally presented in the form of an "HTML" page or an "applet" transmitted to the browser 10.

The second access method, or indirect access method, is in reality also a direct access method, but is implemented from the terminal 1 and not from the smart card 2*a*. This method is essentially used to instantiate virtual objects *Obv<sub>i</sub>* of the dynamic type.

According to this variant of the method, in response to a "URL" that designates a virtual elementary file *fe#x*, the intelligent agent "ATSDA/SGFV" 7 transmits to the browser 10 an "HTML" page that contains a hyperlink that points to the direct access method associated with the virtual object *Obv<sub>i</sub>*.

There are two variants that can be implemented.

The first variant consists of using an "applet." The link to the access method in this case is an "applet" located at the address "@card," which itself can be designated by:

- the name (i.e., a "URL") of a non-virtual file stored in the smart card 2*a*;
- a "URL" that designates a virtual direct file.

A call parameter of this "applet" is a call structure list, for example coded in ASN.1 as indicated above. The "applet" contained in an "HTML" page is downloaded from the smart card 2*a* or the internet network *RI* to the browser 10, which is forced to execute it. This "applet" establishes a session with a first intelligent agent, arbitrarily referenced 232*a<sub>p</sub>*. The connection to this intelligent agent 232*a<sub>p</sub>* uses, for example, a data exchange model of the "TCP/IP" client/server type (i.e., the class known as "socket JAVA"). The "applet" acts as a "TCP/IP" client and connects to a "TCP/IP" server (the latter also being an intelligent agent) identified by the address of the card and a port: "@card:port".

Fig. 13 schematically illustrates the various phases of the exchanges for instantiating a virtual object using the indirect method. This Fig. 13 includes the parameters of the example described above, i.e., the virtual elementary file *fe#5*, which is translated into the "URL" address with the configuration (6) above. It is assumed that the address of the smart card to be used is "@card" and the port 8080. The request is transmitted to the intelligent agent "ATSDA/SGFV" 7 according to the process described above. The latter returns to the browser 10 an "HTML" page *P* constituted by an "applet." In order to simplify the drawing,



the various instructions of this "applet" have been summarized in Fig. 13 by the notation "Applet code" placed between the markers <applet...> and </applet>. In an essentially known way, the "applet" is associated with a "Java" class, which is arbitrarily called "vt.class", for "virtual terminal." The code also includes instructions indicating the address of the first intelligent agent of the list structure, referenced  $232a_p$ , and the address of the port to use, in this case the address "@card" and the port 8081. It should be noted that this intelligent agent  $232a_p$  can be located in the smart card 2a or in the terminal 1.

The next phase consists, for the browser, of requesting the applet from the card 2a using the call structure list, which defines the call parameters of the applet. In response, the card transmits it the applet, which will be loaded by the browser into its virtual "Java" machine, where it will be executed. For the browser, the next phase consists of calling the intelligent agent  $232a_p$  using the "socket" class of the "Java" language.

Each intelligent agent, for example  $232a_p$ , executes a precise task: decryption of an encrypted message, verification of passwords and/or security data, conversion of a file from a first format to another one, etc. Although only one intelligent agent  $232a_p$  is represented, it is possible to provide, as necessary, several cascaded intelligent agents as in the preceding case (Fig. 12), which is illustrated in Fig. 13 by dotted lines. Also as before, each intelligent agent  $232a_p$  consumes a part of the list structure that is addressed to it, and transmits the rest, either unchanged or modified, to the next intelligent agent (not represented).

To better illustrate the first variant, and to illustrate the concept, let us assume that the user  $U_i$  wants to download and execute an audio file, for example coded in the "MP3" format. This file constitutes one of the virtual objects, here referenced  $FS$ , offered by the "HTML" menu page transmitted by the intelligent agent "ATSDA/SGFV" 7 during the initial phase. Fig. 14 schematically illustrates the sequence of steps for instantiating such a virtual object, referenced  $FS$ . It is assumed that the browser 10 does not have an appropriate reader for such a format. This reader, referenced  $LS$ , is searched for on an Internet site, which may or may not be different from the site where the sound file  $FS$  is located.

In the example described, the sequence of steps is as follows:

- a/ the user  $U_i$  clicks on a hyperlink (text, icon or any other graphical representation of the object to be searched for, i.e. the file  $FS$ ): a request  $I_1$  is transmitted to the smart card 2a;
- b/ in response,  $R_1$ , an "HTML" page is transmitted by the smart card 2a to the terminal 1 and to the browser 10;

c/ the "HTML" page received forces the browser 10 to request an "applet":  
interrogation  $I_2$  (in the present case, this means searching for the appropriate sound reader  $LS$ );

d/ in response,  $R_2$ , the reader sought  $LS$  is downloaded and installed in the terminal 1;

e/ the browser 10 again addresses the smart card  $2a$ , request  $I_3$ , in order to obtain an instance of the audio file  $FS$ ; and

f/ in response, the browser 10 receives this audio file  $FS$ , which can be read, i.e. played, by the terminal 1, which now has the appropriate sound reader  $LS$ .

It must be noted that all the operations are transparent for the user  $U_i$ , more precisely for the browser 10, which "knows" only the smart card  $2a$ . The reader  $LS$  (or more generally another "applet") and/or the virtual object sought, i.e. the file  $FS$  in the example, had their sizes been compatible with the storage capacity of the smart card  $2a$ , could then have been stored in the latter (loopbacks symbolized by dotted lines in Fig. 14). The browser 10 does not know the exact location of the virtual objects  $Obv_i$ . Only the smart card  $2a$ , or more precisely the intelligent agent "ATSDA/SGFV" 7, knows the location of the virtual objects of the list of the "SGFV" 8 and the method for accessing them.

In a preferred variant of the method, the intelligent agent "ATSDA/SGFV" 7 also knows the list of the only virtual objects accessible to a given user  $U_i$  (authorizations). It is therefore a secure system. The term "secure" should be considered in its broadest sense. For example, it relates to a payment card that gives access to certain resources, based on a given subscription for example, or cards that provide actual secure access to confidential resources, based on a level of clearance, for example. As indicated, the resources or virtual objects  $Obv_i$  can be constituted by transactions.

This constitutes a characteristic of the method according to the invention.

According to a second variant, illustrated schematically in Fig. 15, it is possible to use a hyperlink that defines the "TCP/IP" address of the first intelligent agent associated with the access method. The address is of the type: "@Agent:AgentPort", with "@Agent" being the actual address of the intelligent agent in question and "AgentPort" being the port of the latter. The list "MethodPDU" in this case is a parameter of a "URL." The hyperlink is associated, for example, with an image or a form of an "HTML" page P'.

Thus, for example, a "URL" having the following structure:

<http://@Agent:AgentPort/MethodPDU?Value-xx...> (7),

makes it possible to reach an intelligent agent that acts as a "TCP/IP web" server, arbitrarily referenced 232a<sub>q</sub>. This intelligent agent 232a<sub>q</sub> is located at the address Agent:AgentPort" and receives the call structure list "MethodPDU" with the parameter "Value=xx...".

To illustrate the concept, let us assume that the virtual object *Obv<sub>i</sub>* is an image to be displayed on a screen (Fig. 1A:5) in a particular format using the browser 10, and that the latter does not have an appropriate program for this display, generally called a "viewer." This could be, for example, a program executable by the operating system used in the terminal 1, of the "XXX.exe" type, with "XXX" being the name of the program. The action of clicking on the above hyperlink (7) will make it possible to search for this executable program, which can be located in the terminal 1 or in a remote system.

The difference between the two variants of embodiment is that in the first case, the browser is "forced" to request the loading of an "applet." All of the steps are performed automatically. In the second case, the user *U<sub>i</sub>* is prompted to click on a hyperlink or to execute a similar action.

Through the reading of the above, it is easy to see that the invention clearly achieves the objects set forth.

It must be clear, however, that the invention is not limited to just the exemplary embodiments explicitly described, particularly in connection with Figs. 2 through 15.

In particular, as concerns the other script translating intelligent agents, the function of the intelligent agent associated with the virtual file management system can be fulfilled by a non-dedicated agent: the "web" agent or the "APDU" agent.

## CLAIMS

1  
2  
3 1. Embedded system, equipped with a chip comprising information processing  
4 means and information storage means and designed to cooperate with a network through a  
5 terminal, characterized in that:

6 - it stores at least one object file containing information associated with an object located  
7 in the network and making it possible to create an instance of this object;

8 - it comprises network interface means designed to cooperate with matching network  
9 interface means located in the terminal, so that the embedded system constitutes an information  
10 server in the network; and

11 - it comprises object file interface means, designed to establish a correspondence between  
12 information passing through the network interface means and assigned to at least said object file,  
13 and information exchanged with said object file.

1 2. Embedded system according to claim 1, wherein the object file comprises a piece  
2 of autonomous software executable in browser software.

1 3. Embedded system according to claim 1, wherein said network interface means are  
2 designed to cooperate with the matching network interface means located in the terminal, so that  
3 the embedded system behaves like a client capable of connecting to at least one server of the  
4 network.

1 4. Method for instantiating an object located in a network, characterized in that it  
2 uses an embedded system, equipped with a chip comprising information processing means and  
3 information storage means, and designed to cooperate with a network through a terminal, the  
4 embedded system storing at least one object file containing information associated with an object  
5 located in the network and making it possible to create an instance of this object, and comprising  
6 network interface means designed to cooperate with matching network interface means located  
7 in the terminal, so that the embedded system constitutes an information server in the network,  
8 and object file interface means designed to establish a correspondence between information  
9 passing through the network interface means and assigned to at least said object file, and

1 information exchanged with said object file, the method also being characterized in that it makes  
2 it possible to describe a set of sessions between agents using an object file, by means of at least  
3 the following steps:

- 4 - establishing a list of the agents implemented;
- 5 - for each agent, defining call arguments necessary to the agent.

1 5. Method according to claim 4, wherein a call argument describes the opening of a  
2 session with another agent.

1 6. Method according to claim 4, wherein an agent modifies the list of arguments  
2 used by another agent.

1 7. Method for instantiating an object located in a network, characterized in that it  
2 uses an embedded system, equipped with a chip comprising information processing means and  
3 information storage means and designed to cooperate with a network through a terminal, the  
4 embedded system storing at least one object file containing information associated with an object  
5 located in the network and making it possible to create an instance of this object, and comprising  
6 network interface means designed to cooperate with matching network interface means located  
7 in the terminal, so that the embedded system constitutes an information server in the network,  
8 and object file interface means designed to establish a correspondence between information  
9 passing through the network interface means and assigned to at least said object file and  
10 information exchanged with said object file, the method also being characterized in that it  
11 implements sessions between agents described by an object file executed from the information  
12 server of the embedded system by means of at least the following steps:

- 13 - identification of an object file;
- 14 - execution of this object file.

1 8. Method according to claim 7, wherein the object file is executed by instantiating  
2 the first agent associated with the object file.

9. Method according to claim 7, wherein the object file is executed by instantiating one or more agents referenced by the object file.

10. Method for instantiating an object located in a network, characterized in that it uses an embedded system, equipped with a chip comprising information processing means and information storage means and designed to cooperate with a network through a terminal, the embedded system storing at least one object file containing information associated with an object located in a network and making it possible to create an instance of this object, and comprising network interface means designed to cooperate with matching network interface means located in the terminal, so that the embedded system constitutes an information server in the network, and object file interface means designed to establish a correspondence between information passing through the network interface means and assigned to at least said object file and information exchanged with said object file, the method also being characterized in that it implements sessions between agents described by an object file executed from browser software by means of at least the following steps:

- loading by the browser software of an object file and a specific software capable of implementing it;
- execution of the specific software by the browser software.

11. Method according to claim 10, wherein the specific software is embodied in any interpreted language executable by the browser software.

12. Method according to claim 10, wherein the object file interpreter is embodied in browser software.

13. Method for instantiating an object located in a network, characterized in that it uses an embedded system, equipped with a chip comprising information processing means and information storage means and designed to cooperate with a network through a terminal, the embedded system storing at least one object file containing information associated with an object located in the network and making it possible to create an instance of this object, and comprising network interface means designed to cooperate with matching network interface means located

1 in the terminal, so that the embedded system constitutes an information server in the network,  
2 and object file interface means designed to establish a correspondence between information  
3 passing through the network interface means and assigned to at least said object file, and  
4 information exchanged with said object file, the method also being characterized in that it  
5 enables the embedded system to make it possible to implement sessions between agents  
6 described by an object file executed from browser software, and in that it comprises the step that  
7 consists of identifying, by means of a universal resource identifier, a specific software  
8 implementing the browser software.

1 14. Method according to claim 13, wherein the universal resource identifier is  
2 integrated into a hypertext document.

1 15. Method according to claim 13, wherein said specific software is loaded by a  
2 method available in the browser software and deduced from the universal resource identifier.

## ABSTRACT

The invention relates to a method and an architecture for securely accessing virtual objects ( $Obv_i$ ) distributed in systems connected to the internet network ( $RI$ ), and for obtaining an instance of same. This access is performed via a smart card ( $2a$ ), through a "web" browser (10).

5 The terminal (1) and the smart card ( $2a$ ) each comprise a specific protocol layer (13,  $23a$ ). The latter comprises intelligent agents ( $132, 232a_i$ ) for establishing two-way data exchange sessions, thereby allowing the smart card ( $2a$ ) to have a "web" server functionality. The smart card ( $2a$ ) also comprises intelligent agents, called script translators, and a virtual file management system (8) cooperating with a specialized script-translating intelligent agent (7). Each virtual object ( $Obv_i$ ) is associated with a virtual file of the virtual file management system (8). The specialized

10 intelligent agent (7) presents the browser (10) with a list of the accessible virtual objects ( $Obv_i$ ) and generates methods for accessing these objects.

FIG. 8

#9124645-US3815/BC-Literal Translation of PCT APP.



1/8

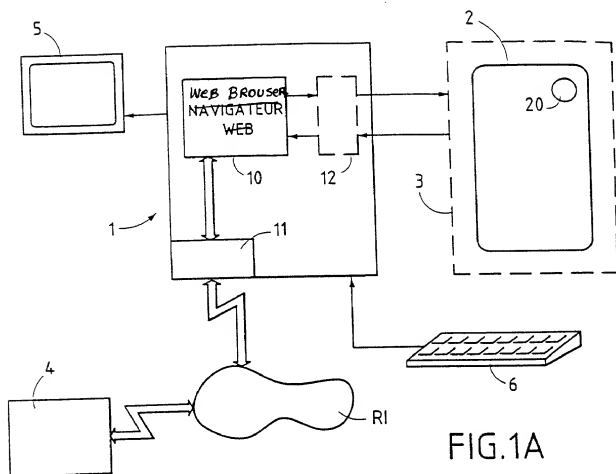


FIG. 1A

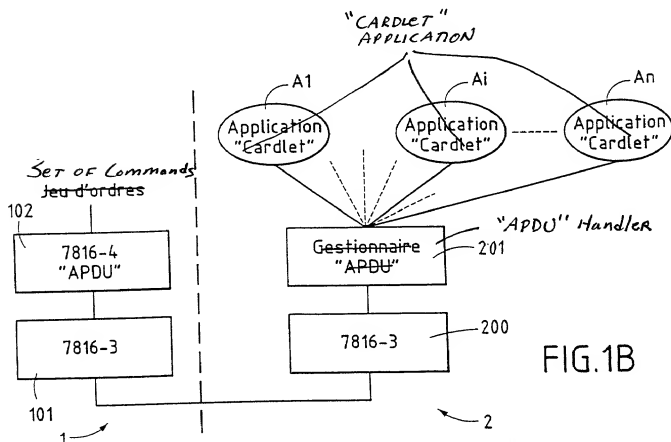


FIG. 1B

FIG. 2

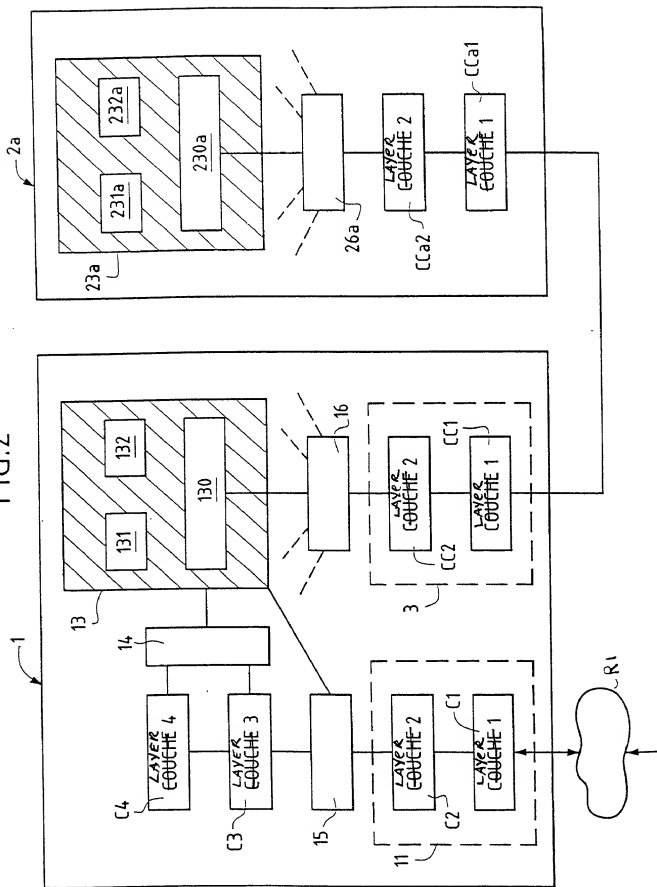
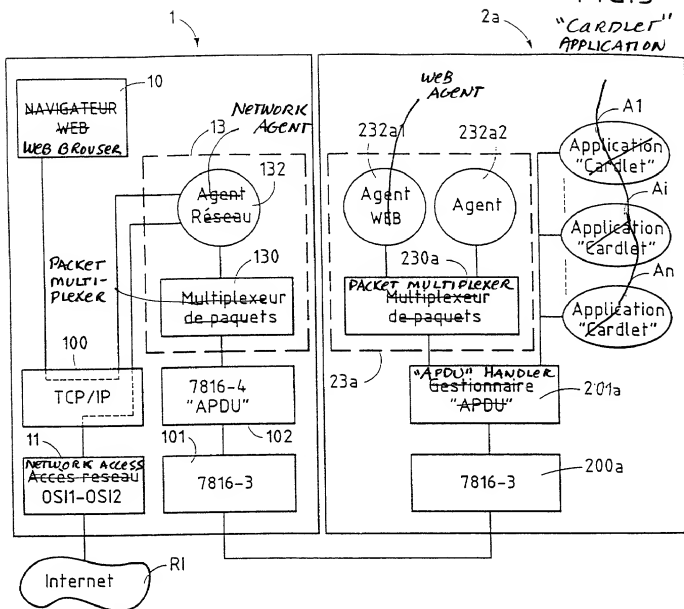
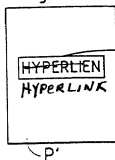


FIG.3



Html Page  
Page Html



<http://@Agent:AgentPort/MethodPDU?value=xx>



FIG. 15  
1<sup>er</sup> Agent De la Liste 1st Agent on the LIST OF "MethodPDU"  
de Structure d'Appel "MethodPDU" Call Structures

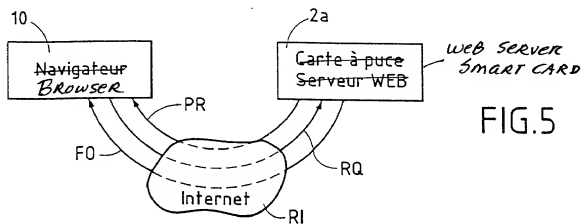
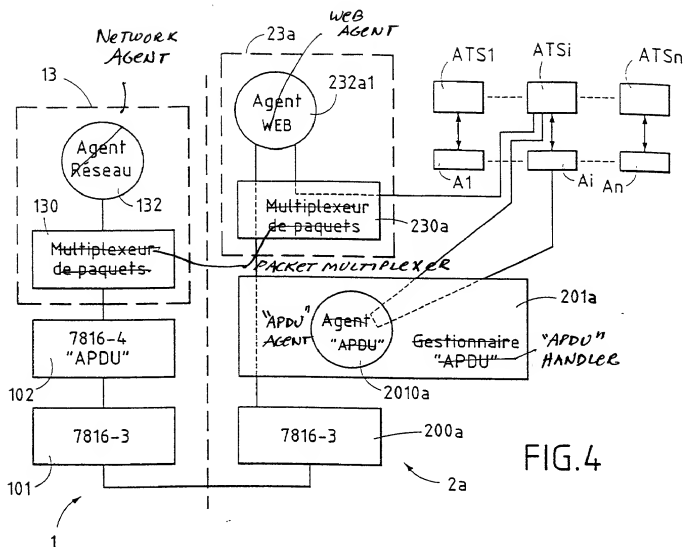


FIG.6

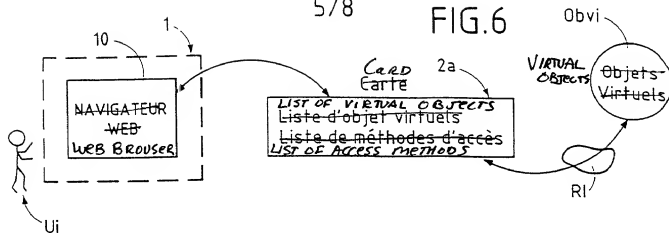


FIG.7

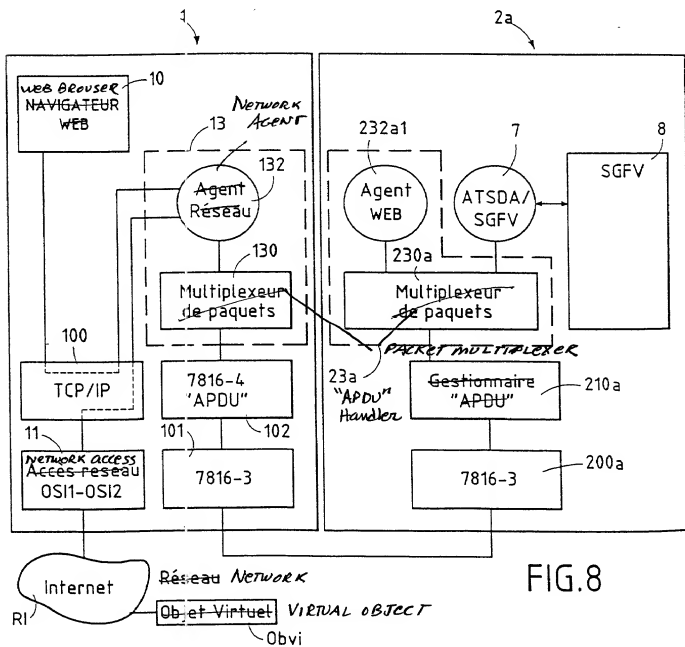
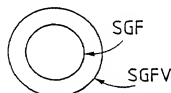
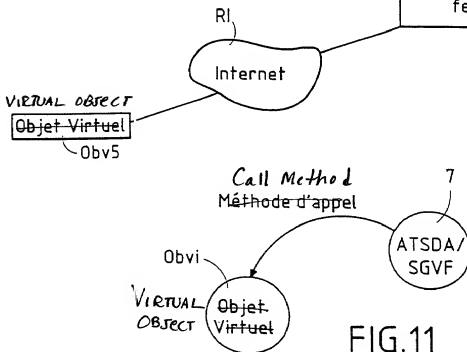
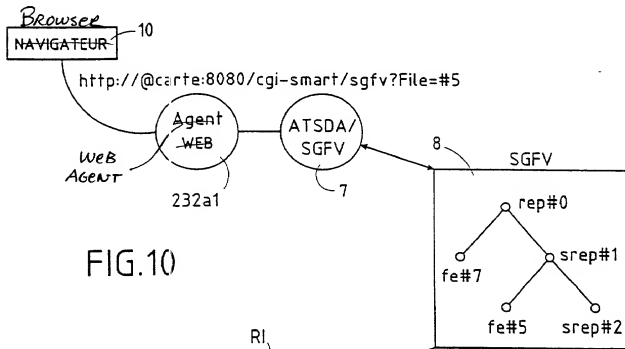
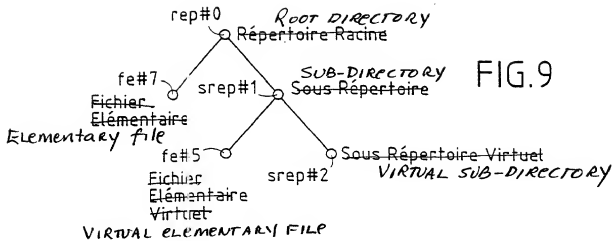


FIG.8

6/8



7/8

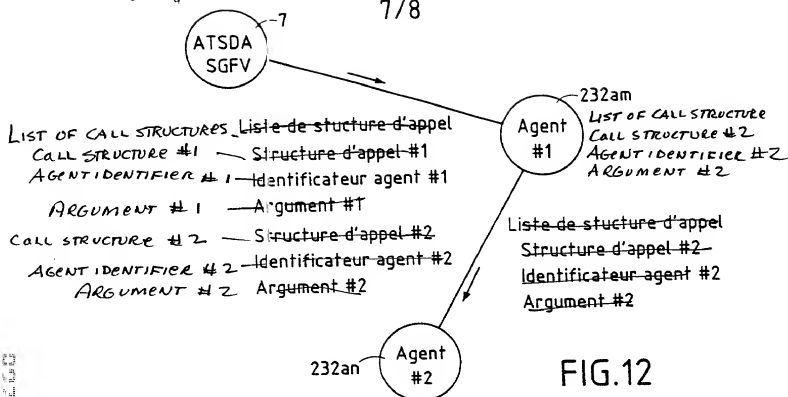


FIG. 12

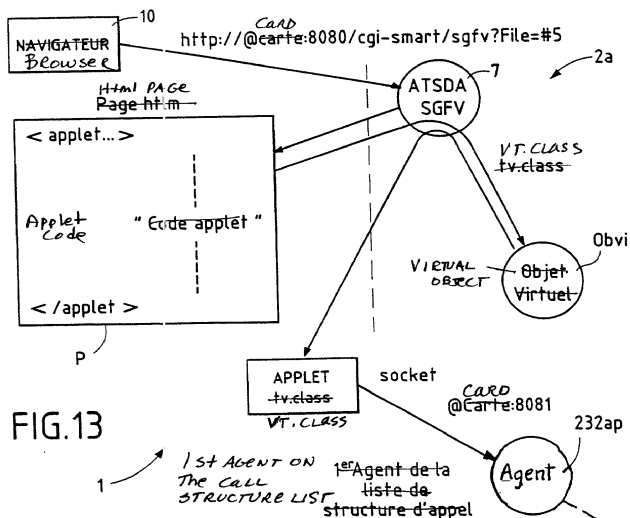


FIG. 13

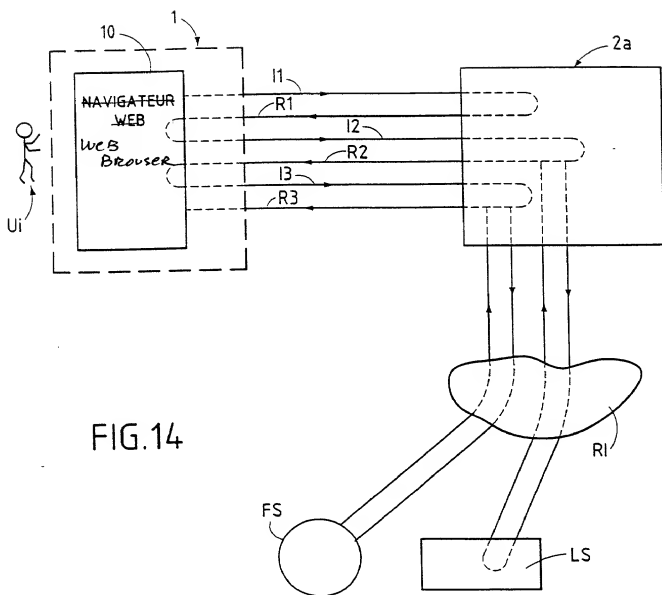


FIG.14



# Declaration and Power of Attorney For Patent Application

## Declaration Pour Demandes de Brevets Avec Pouvoirs

### French Language Declaration

En tant qu'inventeur nommé ci-après, Je déclare par le présent acte que:

Mon nom, mon domicile, mon adresse postale, ma nationalité sont ceux qui figurent ci-après,

Je déclare que je crois être l'inventeur original, premier et unique (si un seul nom figure sur le présent acte) ou un des co-inventeurs, originaux et premiers (si plusieurs noms figurent sur le présent acte) du sujet revendiqué et pour lequel un brevet est demandé sur la base de l'invention intitulée:

Procédé d'accès à un objet à l'aide d'un  
navigateur de type "Web" coopérant avec une  
carte à puce et architecture pour la mise en  
œuvre du procédé.

dont la description  
(cocher la case correspondante)

☒ est annexée au présent acte.

☐ a été déposée \_\_\_\_\_

Numéro de série de la demande \_\_\_\_\_

et modifiée le \_\_\_\_\_  
(si approprié)

Je déclare par le présent acte avoir examiné et compris le contenu de la description identifiée ci-dessus, revendications y compris, et le cas échéant telle que modifiée par l'amendement cité plus haut.

Je reconnais le devoir de divulguer l'information qui est en rapport avec l'examen de cette demande selon Titre 37 du Code des Règlements Fédéraux §1.56(a).

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

the specification of which  
(check one)

☐ is attached hereto.

☐ was filed on \_\_\_\_\_ as

Application Serial No. \_\_\_\_\_

and was amended on \_\_\_\_\_  
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

## French Language Declaration

Je revendique par le présent acte le bénéfice de priorité étrangère selon Titre 35, du Code des Etats-Unis, §119 de toute demande de brevet ou d'attestation d'inventeur énumérée ci-après, et j'ai identifié également ci-après toute demande étrangère de brevet ou d'attestation d'inventeur ayant une date de dépôt antérieure à celle de la demande pour laquelle la priorité est revendiquée.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior foreign applications

Demande(s) de brevet antérieure(s) dans un autre pays:

FR 9903172

France

15 03 1999

(Number)  
(Numero)

(Country)  
(Pays)

(Day/Month/Year Filed)  
(Jour/Mois/Année de dépôt)

Priority claimed

Droit de priorité  
revendiqué

☒

Yes

Oui

☐

No

Non

(Number)  
(Numero)

(Country)  
(Pays)

(Day/Month/Year Filed)  
(Jour/Mois/Année de dépôt)

☐

Yes

Oui

☐

No

Non

(Number)  
(Numero)

(Country)  
(Pays)

(Day/Month/Year Filed)  
(Jour/Mois/Année de dépôt)

☐

Yes

Oui

☐

No

Non

Je revendique par le présent acte, le bénéfice selon Titre 35 du Code des Etats-Unis, §120 de toute(s) demande(s) américaines énumérée(s) ci-après et, dans la mesure où le sujet de chacune des revendications de cette demande n'est pas divulgué dans la demande américaine antérieure, de la façon définie par le premier paragraphe de Titre 35 du Code des Etats-Unis, §112, je reconnais le devoir de divulguer l'information pertinente selon Titre 37 du Code des Règlements Fédéraux, §1.56(a), toute information qui se présente entre la date de dépôt de la demande antérieure et la date de dépôt de la demande, soit nationale, soit internationale PCT.

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial No.)  
(No. de Demande)

(Filing Date)  
(Date de Dépôt)

(Etat)  
(brevetée, pendante,  
abandonnée)

(Status)  
(patented, pending,  
abandoned)

(Application Serial No.)  
(No. de Demande)

(Filing Date)  
(Date de Dépôt)

(Etat)  
(brevetée, pendante,  
abandonnée)

(Status)  
(patented, pending,  
abandoned)

Je déclare par le présent acte que toutes mes déclarations, à ma connaissance, sont vraies et que toutes les déclarations faites à partir de renseignements ou de suppositions, sont tenues pour être vraies; de plus, toutes ces déclarations ont été faites en sachant que de fausses déclarations volontaires ou autres actes de même nature sont sanctionnés par une amende ou un emprisonnement, ou les deux, selon la Section 1001, du Titre 18 de Code des Etats-Unis et que de telles déclarations délibérément fausses peuvent compromettre la validité de la demande ou du brevet délivré.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

## French Language Declaration

POUVOIR: En tant qu'inventeur, je désigne l'(les) avocat(s) et/ou l'(les) agent(s) suivant(s) pour poursuivre la procédure de cette demande et traiter toute affaire la concernant supris du Bureau des Brevets et de Marques:

Harold L. Stowell, Reg. 17,233  
 Edward J. Kondracki, Reg. 20,604  
 Dennis P. Clarke, Reg. 22,549  
 William L. Feeney, Reg. 29,918  
 John C. Kerins, Reg. 32,421

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (list name and registration number)

Harold L. Stowell, Reg. 17,233  
 Edward J. Kondracki, Reg. 20,604  
 Dennis P. Clarke, Reg. 22,549  
 William L. Feeney, Reg. 29,918  
 John C. Kerins, Reg. 32,421

Adresser toute correspondance à:

Edward J. Kondracki, Esq.  
 KERMAM, STOWELL, KONDRACKI  
 & CLARKE, P.C.  
 5203 Leesburg Pike, Suite 600  
 Falls Church, VA 22041

Send Correspondence to:

Edward J. Kondracki, Esq.  
 KERMAM, STOWELL, KONDRACKI  
 & CLARKE, P.C.  
 5203 Leesburg Pike, Suite 600  
 Falls Church, VA 22041

Adresser toute communication téléphonique à:  
 (Nom) (Numéro de téléphone)

Edward J. Kondracki, Esq.  
 (703) 998-3302

Direct Telephone Calls to: (name and telephone number)

Edward J. Kondracki, Esq.  
 (703) 998-3302

Nom complet du seul ou premier inventeur		Full name of sole or first inventor	
URIEN Pascal			
Signature de l'inventeur	Date	Inventor's signature	Date
<i>[Signature]</i>	31-03-99		
Domicile		Residence	
4 rue du Ruisseau St Prix 78450 VILLEPREUX FRANCE			
Nationalité		Citizenship	
Française		FRX	
Adresse Postale		Post Office Address	
4 rue du Ruisseau St Prix 78450 VILLEPREUX FRANCE			
Nom complet du second co-inventeur, le cas échéant		Full name of second joint inventor, if any	
Signature de l'inventeur		Second Inventor's signature	
Date		Date	
Domicile		Residence	
Nationalité		Citizenship	
Adresse Postale		Post Office Address	

(Fournir les mêmes renseignements et la signature de tout co-inventeur supplémentaire.)

(Supply similar information and signature for third and subsequent joint inventors.)

T2146-906652-US3815/BC

**IN THE UNITED STATES DESIGNATED/ELECTED OFFICE (D.O./E.O./US)**

Applicant: Pascal URIEN

International  
Application No.: PCT/FR00/00625

International  
Filing Date: 15 March 2000

U.S. Serial No.: To be Assigned

U.S. Filing Date: November 15, 2000

For: **SYSTEM FOR ACCESSING AN OBJECT USING A "WEB"  
BROWSER CO-OPERATING WITH A SMART CARD**

**CHANGE OF ADDRESS**

Assistant Commissioner for Patents  
Washington, DC 20231

Sir:

Effective immediately, please note our new correspondence address and  
telephone/fax number as follows:

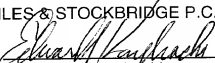
Miles & Stockbridge P.C.  
1751 Pinnacle Drive, Suite 500  
McLean, VA 22102-3833  
Telephone: 703-903-9000  
Fax: 703-610-8686

Respectfully submitted,

MILES & STOCKBRIDGE P.C.

Date: November 15, 2000

By:

  
Edward J. Kondracki  
Registration No. 20,604

1751 Pinnacle Drive – Suite 500  
McLean, VA 22102-3833  
Tel.: 703/903-9000  
Fax: 703/610-8686